

AD-A194 303

RANDOM-ACCESS ALGORITHMS FOR ENVIRONMENTS WITH CAPTURE

1/1

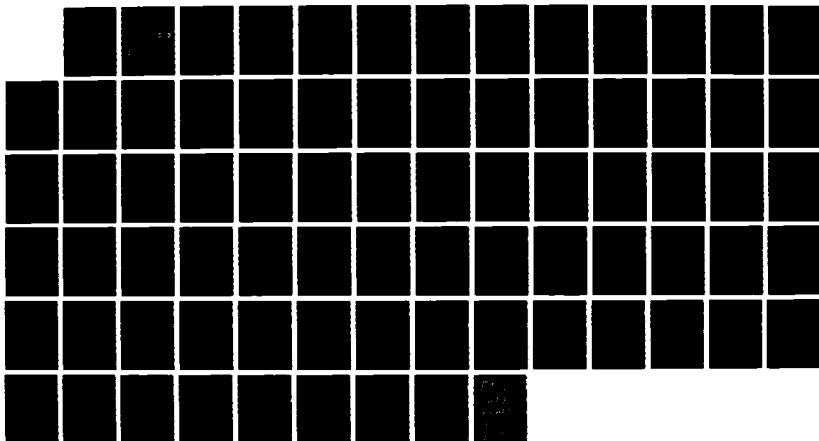
(U) VIRGINIA UNIV CHARLOTTESVILLE DEPT OF ELECTRICAL
ENGINEERING. P PAPANTONI-KAZAKOS ET AL. 19 APR 88

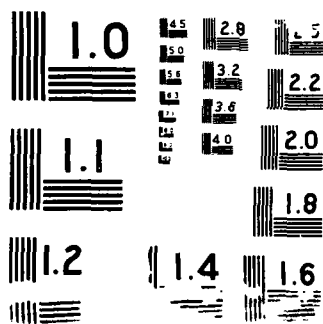
UNCLASSIFIED

UVA/525415/EE88/113 N00014-86-K-0742

F/G 25/2

NL





AD-A194 303

DTIC FILE COPY

4

A Technical Report
Grant No. N00014-86-K-0742
September 1, 1986 - August 31, 1988

RANDOM-ACCESS ALGORITHMS FOR ENVIRONMENTS WITH CAPTURE

Submitted to:

Office of Naval Research
Department of the Navy
800 N. Quincy Street
Arlington, Virginia 22217-5000

Submitted by:

P. Papantoni-Kazakos
Professor

D. F. Lyons
Graduate Assistant

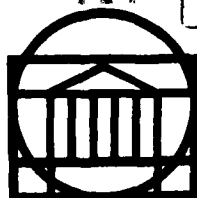
DTIC
ELECTE
APR 26 1988
S H D

Report No. UVA/525415/EE88/113

April 1988

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited



SCHOOL OF ENGINEERING AND
APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA 22901

88 4 26 09 6

A Technical Report
Grant No. N00014-86-K-0742
September 1, 1986 - August 31, 1988

RANDOM-ACCESS ALGORITHMS FOR ENVIRONMENTS WITH CAPTURE

Submitted to:

Office of Naval Research
Department of the Navy
800 N. Quincy Street
Arlington, Virginia 22217-5000

Submitted by:

P. Papantoni-Kazakos
Professor

D. F. Lyons
Graduate Assistant

Department of Electrical Engineering
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/525415/EE88/113
April 1988

Copy No. _____

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UVA/525415/EE88/113			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION University of Virginia Dept. of Electrical Engr.		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research Resident Representative		
6c. ADDRESS (City, State, and ZIP Code) Thornton Hall Charlottesville, Virginia 22901			7b. ADDRESS (City, State, and ZIP Code) 818 Connecticut Ave., N.W., Eighth Floor Washington, DC 20006		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Office of Naval Research		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0742		
8c. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, Virginia 22217-5000			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Random-Access Algorithms For Environments with Capture					
12. PERSONAL AUTHOR(S) P. Papantoni-Kazakos and D. F. Lyons					
13a. TYPE OF REPORT		13b. TIME COVERED FROM 9/1/86 TO 8/31/88		14. DATE OF REPORT (Year, Month, Day) 1988 April 19	
				15. PAGE COUNT 68	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Communication networks have seen a remarkable rise in both applications and theoretical study in the the past twenty years. An increased demand for flexible and easily modified communication systems has supported recent interest in applications of communication networks to a wide variety of problems such as mobile packet radio, distributed sensor systems, satellite networks, and a plethora of applications in integrated computer communications. Such systems are typically characterized by a large number					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. R. N. Madan			22b. TELEPHONE (Include Area Code)		22c. OFFICE SYMBOL

19. ABSTRACT (cont.)

λ

of users attempting to communicate with a central node via a common communication media. The multiaccess communication problem examines how to utilize most efficiently the available communication resources.

This thesis proposes two random-access algorithms appropriate for operation in networks employing capture (that is, the correct reception of a single transmission in the presence of multiple transmissions). Both the paradigmatic ternary-feedback model as well as an enriched-feedback scheme are considered. After observing the regenerative properties of these algorithms, several results from renewal theory are employed to compute the throughput and delay characteristics of the proposed algorithms. It is shown that significant performance improvements (as compared to non-capture systems) are possible using systems with capture and employing appropriate random-access algorithms. Extensive numerical results are included.

Acknowledgment

I would like to thank my advisor Professor Titsa Papantoni-Kazakos for her help and encouragement. I am also indebted to Michael Paterakis for his patience and help throughout this project. I am grateful to the U.S Office of Naval Research for sponsoring this work under contract N00014-86-K-0742. Finally I thank Professor Steve Wilson for encouraging me to pursue graduate study at the University of Virginia.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Table of Contents

Chapter 1 Introduction	1
1.1 Network System Model	1
 Chapter 2 System Models for the Capture Environment	 3
2.1 System Designs to Create Capture	6
2.2 Systems Where Capture is Imposed	6
2.3 Generalized Capture Model	9
 Chapter 3 An Algorithm for Enriched Feedback	 9
3.1 Description of Algorithm for Enriched Feedback	12
3.2 Algorithm Analysis	12
3.3 The Capetanakis Algorithm in the Capture Environment	16
3.4 Conclusions for the MTCWA	25
 Chapter 4 An Algorithm for Simple Ternary Feedback	 31
4.1 Description of Algorithm for Simple Ternary Feedback	32
4.2 Analysis of Algorithm	32
4.3 Numerical Results, Conclusions, and Comparisons	35
 Chapter 5 Conclusions	 40
5.1 The MTCWA Compared to the Capetanakis Dynamic Algorithm	50
5.2 Random-Access Networks with Capture	50
5.3 Algorithm Sensitivity to Forward-Channel Errors	50
5.4 System Costs to Achieve Capture	51
 References	 52
Appendix I Analysis of Algorithm for Enriched Feedback	53
Appendix II Analysis of Algorithm for Simple Ternary Feedback	57

CHAPTER 1

Introduction

Communication networks have seen a remarkable rise in both applications and theoretical study in the the past twenty years. An increased demand for flexible and easily modified communication systems has supported recent interest in applications of communication networks to a wide variety of problems such as mobile packet radio, distributed sensor systems, satellite networks, and a plethora of applications in integrated computer communications.

Typical communication system analysis assumes the existence of a point-to-point link between two "nodes" desiring to exchange information. The received signal at either end of the link, therefore, depends only on the available bandwidth and the noise process corrupting the communication media. However, in the network environment considered herein many users share a common channel, and attempt to communicate with a central node. Therefore, in communication networks such as multitap bus systems or packet radio networks the received signal at any one node may be affected by the signals being transmitted at any of the other nodes in the network. The received signal at any one node may, thus, be modeled as a sum of attenuated and delayed transmissions from several sources and an additive noise component. The multiaccess communication problem is how to utilize best the available communication media.

Many references ([1], [2]) discuss the basic design considerations and trade-offs inherent to the selection of a multiaccess technique. The usual figures of merit for a communication network are the throughput (which is approximately the probability of a slot being used for a successful transmission) and the average delay experienced by a transmission. The stochastic nature of the user population employing the network is of paramount importance in selecting a multiaccess technique. For a small number of high-duty-cycle users a network is best established using a deterministic scheduling scheme such as time-division multiplexing (TDM) or frequency-division multiplexing (FDM). Such deterministic systems with n users can achieve throughputs very close to 1 with average delay on the order of $n/2$ units of time. The more difficult and interesting problem occurs with a large (possibly time-varying) number users, which generate traffic in a bursty manner. Deterministic techniques may still provide excellent throughput for such populations; however, average delay is still on the order of $n/2$ units of time, which may be very large for n large. For such situations random-access techniques may be employed to lower delay with a corresponding reduction in throughput for the network. In this report two specialized random-access algorithms appropriate for a specific environment (that is, the capture environment) are designed and analyzed. For concreteness the usual assumptions implicit in the analysis of a random-access network are discussed. Different modifications are then discussed and justified for such networks and random-access algorithms are presented for the modified environments.

1.1 Network System Model

In order to focus attention on the multiaccess aspects of networks, a somewhat idealized model of an actual network is assumed.

The following assumptions are paradigmatic for theoretical studies of random-access techniques.

- (1) The entire network is assumed to be synchronized to a common, slotted channel. Channel time is partitioned into slots so that time may be measured in integers on the channel. Users transmit packets of data with duration equal to one channel slot. All packet transmissions are restricted to beginning at slot boundaries. For many networks the use of a slotted channel can significantly increase performance [2]; furthermore, this assumption allows for simpler discrete mathematics to be used in the analysis of the system.
- (2) The aggregate arrival process from the user population to the network is Poisson distributed. The number of users generating packets for transmission in any one channel slot is, therefore, a Poisson random variable. This assumption has been shown [3] provide lower bounds on performance among models for independent and identically distributed users. The mean arrival rate to the system is given by the Poisson parameter (typically represented as λ) and has units of packets/slot.
- (3) Transmissions within the network are received correctly with probability 1 if only one transmission is attempted in the network during a specific time slot. If two or more users transmit during the same time slot all information is assumed lost and all

users must retransmit their packets. This model is sometimes referred to as the perfect reception/collision model. A modification to the usual reception/collision model is considered below.

- (4) Errorless, instantaneous feedback describing the outcome of the current slot is assumed available to all users in the network. Ternary feedback is often used which allows for all users to identify successful slots (S), collisions (C), or empty slots (E). Binary (collision or non-collision) feedback is also sometimes employed at the expense of a reduction in performance. Appropriate modifications to the feedback for the networks considered herein will be presented. In particular the effect of augmenting the feedback with the identity of the successful packet will be considered.
- (5) All users are assumed to be full sensing. This implies that all users in the network have been observing the network operations and feedback since the network began operating. This might appear initially as a rather unrealistic assumption. However, protocols may be designed so that new users to the system can become synchronized to the algorithm operations after a short delay.
- (6) All packets are assumed successfully transmitted after some delay. This assumption implies that packets involved in collisions are retransmitted until they are successful. For many communication networks this might seem an obvious and strict requirement; however, some networks such as those collecting sensor data may not require all packets generated to be successfully transmitted. Instead a

lower limit might be imposed on the fraction of successfully transmitted packets. This possibility is explored herein.

For the idealized network postulated above a multitude of random-access algorithms have been presented and analyzed (for example [4], [5], [6]). The primary import of this work is to analyze algorithms appropriate for network environments with capture. Capture occurs when one packet is successfully received in a slot wherein two or more users attempted transmission. Note this relaxes the rather pessimistic assumption in (3) that no information is exchanged in the presence of multiple transmissions. In this paper capture is modeled as a probabilistic mechanism. Deterministic or perfect capture has also been considered (for example, [15]). Specific features of the capture environment and some brief justifications are presented in Chapter 2. Chapters 3 and 4 describe and analyze two random-access algorithms for different environments with capture. The basic analytical framework and systems of equations used to derive the throughput and delays for these algorithms are presented in Appendices I and II. Chapter 5 presents conclusions on the results obtained in the previous chapters.

CHAPTER 2

System Models for the Capture Environment

This chapter presents the details of the capture environment introduced in Chapter 1. Systems with capture are presented from two viewpoints. First, capture is viewed as a system attribute created via system design. This may be seen as a more sophisticated system than that described in Chapter 1 and it is seen that with the addition of enriched feedback such a system is capable of high-throughput, low-delay performance. Second, capture is assumed to be a system characteristic that is imposed on the system due to other design requirements for the system. This might be seen in a network suitable for strategic scenarios. A simple algorithm that operates with only ternary feedback is investigated for this environment.

2.1 System Designs to Create Capture

As will be demonstrated in Chapter 3, communication networks that can exploit capture are capable of significantly better performance than networks without capture. However, significant increases in the complexity and capacity of the feedback channel are required. Several authors have investigated random-access algorithms with capture and enriched feedback, and have reported significant performance improvements [7], [8], [9]. It is, therefore, reasonable to design a more expensive system that allows for capture. Certain spread-spectrum encoding techniques are candidates for capture

systems since transmissions received with slight time-delay differences might be decoded separately. Some authors [7] have proposed a technique wherein each user randomly selects its transmission power according to some prescribed rule. Thus, users transmitting at higher power levels in any one slot are more likely to be captured than lower power transmissions. However, in many networks with large numbers of users transmission power is an expensive commodity. Requiring each user to use a wide range of transmission powers (90 dB in [7]), therefore, seems impractical.

A method of achieving capture via time randomization proposed by Davis and Gronemeyer [8] appears as a practical method of achieving capture. In their scheme a small time-randomization interval is added to the beginning of each slot. Thus, slots are slightly longer in duration than packets; however, significant increases in system throughput per slot can more than compensate for this small amount of overhead. A user desiring to transmit in any slot then chooses a random starting time for his packet according to a uniform distribution on $[0, t_r]$. If the first two transmissions in any given slot are separated by more than t_c , where t_c is a system design parameter, then the first packet transmitted is successfully received and capture occurs. The other users transmitting in the capture slot are not captured and must, therefore, retransmit their packets in subsequent slots. Figure 2.1 shows an example slot for this scheme wherein two users attempt transmission. In this example if $x \geq t_c$, packet 1 is captured. For this system the probability of capture conditioned on k users transmitting may be found to be

$$P_k = \left[\frac{t_r - t_c}{t_r} \right]^{k-1} \quad (2.1)$$

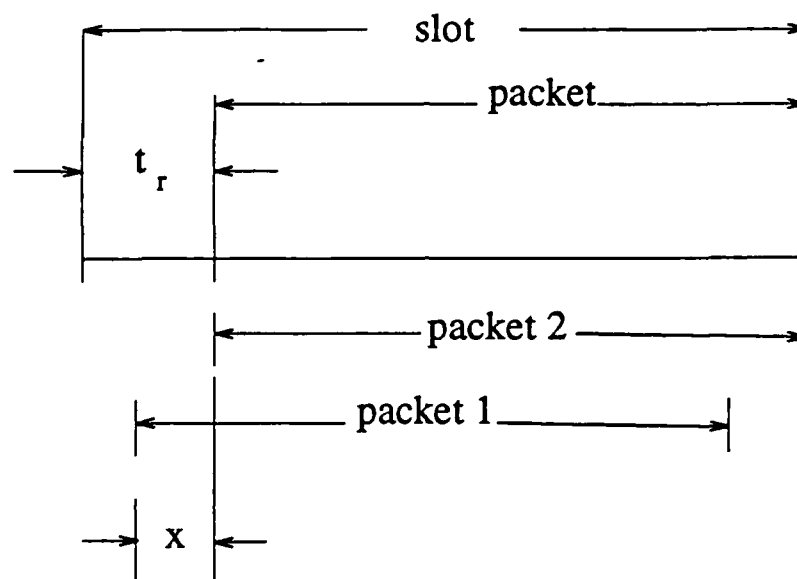


Figure 2.1

Example Slot for Time-Randomization Capture

The above expression indicates systems that can capture a packet with only a small time offset from the following packets (that is, t_c small) can achieve capture with high probability in low-multiplicity collisions. The randomization interval, t_r , could also be increased; however, this lowers the overall capacity of the system since a larger amount of time is spent transmitting each packet.

2.2 Systems Where Capture is Imposed

In some packet radio systems capture may be an effect imposed by other system design considerations. An example of such an environment is a mobile packet radio net that is subject to severe Rayleigh fading and/or other interference. In such circumstances the communication media is time varying; thus, different users would be more likely to be captured during different time intervals. In the presence of multiple transmissions one or more of the signals may be subject to such severe fading that they are undetected by the central receiver. In this case a single successful transmission is indistinguishable from a capture in the presence of multiple transmissions. Furthermore, single transmissions may be received correctly with probability less than one in such systems. These considerations imply the design of a simple ternary-feedback algorithm appropriate for operation under stressed conditions.

2.3 Generalized Capture Model

Throughout the analysis performed herein a generalized capture model is assumed where

$$p(\text{capture} | k \text{ simultaneous transmissions}) = pq^{k-1}$$

The set $\{P_k = pq^{k-1}_{k \geq 1}\}$ defined above determines the set of "capture probabilities" for the system, and represents approximately a variety of capture environments. Note that $p=1$ for the case where a single transmission is successfully received with probability 1. The parameter q is a system characteristic. For $p=1$ and $q \rightarrow 1$ the system achieves near perfect capture in any slot where 1 or more users transmits. The case $p=1$ and $q=0$ is the typical perfect reception no-capture system usually considered in analysis of multiaccess systems. This set of capture probabilities is easily parameterized and has the flexibility to represent randomized systems such as in [8] as well as the effects of noise ($p \neq 1$).

Random-access algorithms are developed for two different types of feedback in the capture environment.

- (1) In Chapter 3 a random-access algorithm is developed that assumes the identity of a captured packet is broadcast to the users along with the usual ternary feedback discussed in Chapter 1. Thus, when there are multiple transmissions in a slot and one of the packets is captured, a S feedback as well as the identity of the captured packet is broadcast to all users. This clearly implies an enriched feedback capability since the feedback is no longer ternary. For a system of N users this implies a bandwidth expansion of approximately $\log N$ in the feedback channel. This type of enriched feedback would likely be incorporated into a system designed to create capture as described above.

- (2) In Chapter 4 a random-access algorithm is presented that operates in capture using only ternary feedback. In the design and analysis of this algorithm it is assumed that the receiver cannot distinguish between a single successful transmission and a capture in the presence of multiple transmissions. Moreover in contrast to the environment in (1), users themselves cannot discern the identity of the successful packet. This is, indeed, a very pessimistic system, but one may that may be realistic for stressed or strategic systems as mentioned above.

For each of the above cases appropriate window random-access algorithms are proposed and analyzed. These algorithms are essentially modifications of the algorithm in [10]. For the enriched-feedback case the algorithm proposed takes advantage of the higher-level feedback to provide higher throughputs and low delays as the probability of capture increases. For the case with only ternary feedback it is recognized that a user can never be certain with probability 1 that its packet was successfully received. It is, therefore, necessary to allow for some portion of the traffic input to the algorithm to be lost without ever being successfully transmitted. An algorithm is developed that allows a trade-off between the fraction of lost packets and delays.

CHAPTER 3

An Algorithm for Enriched Feedback

This chapter describes the design and analysis of a random-access algorithm suitable for capture environments and enriched feedback. For comparison a modification of the well-known Capetanakis dynamic algorithm [4] is also considered and analyzed.

3.1 Description of Algorithm for Enriched Feedback

Considering the system model in Chapters 1 and 2, we adopt a modification of the window RAA in [10]. The modification is necessary since when a user transmits in a slot and observes an S feedback, the user decides whether or not its own packet was successful (based on the packet i.d. in the feedback), and whether or not it must retransmit the current packet. We call the algorithm the Modified Two Cell Window Algorithm (MTCWA). We first state its operations; subsequently, we discuss its operational characteristics and its differences from the algorithm in [10].

The MTCWA utilizes a window of length Δ . Let t be a time instant such that, for some $t_1 < t$ all the packet arrivals in $(0, t_1]$ have been successfully transmitted and there is no information regarding the arrival interval $(t_1, t]$, and such that t corresponds to the beginning of some slot. The instant t is called a collision resolution point (CRP), the arrival interval $(0, t_1]$ is called a "resolved interval", and the quantity $t_1 - t$ is called the "lag at t ". In slot t the packet arrivals in $(t_1, t_2 \triangleq \min(t_1 + \Delta, t)]$ attempt transmission, and

the arrival interval $(t_1, t_2]$ is called the "examined interval". The examined interval is resolved when all the arrivals in it have been successfully received by the receiver and this event is known to all users. Until $(t_1, t_2]$ is resolved no arrivals in (t_2, ∞) are allowed transmission. The time period required for the resolution of an examined interval is called the Collision Resolution Interval (CRI). The algorithm rules are as follows:

1. If the examined interval $(t_1, t_2]$ contains zero packets, then the CRI lasts one slot, and a new examined interval $(t_2, t_3 \triangleq \min(t_2 + \Delta, t+1))$ is selected at $t+1$.
2. If the examined interval $(t_1, t_2]$ contains one packet and $x_t = S$, then slot $t+1$ is wasted, with $x_{t+1} = E$, so that it becomes known to all users that the examined interval has been resolved. Thus, the CRI lasts two slots, and a new examined interval is selected at $t+2$.
3. If the examined interval $(t_1, t_2]$ contains at least one packet and $x_t = C$, then the CRI lasts at least three slots. During the time period that the CRI lasts each involved user implements the algorithm rules independently via the use of a counter. Given some user the value of his counter at time t is denoted r_t , where r_t equals either 1 or 2. The utilization and updating of the counter values and the identification of the slot when the CRI ends, are as follows:

3.1 The user transmits in slot t , if and only if $r_t = 1$.

3.2 The counter values are updated as follows:

- (a) If $x_{t-1} = E$ or S and $r_{t-1} = 2$, then $r_t = 1$.
- (b) If $x_{t-1} = C$ and $r_{t-1} = 2$, then $r_t = 2$.
- (c) If $x_{t-1} = S$, $r_{t-1} = 1$, and the user identifies capture for himself, then his captured packet departs the system.
- (d) If $x_{t-1} = S$, $r_{t-1} = 1$, and the user identifies no capture for himself, then $r_t = 1$.
- (e) If $x_{t-1} = C$ and $r_{t-1} = 1$, then

$$r_t = \begin{cases} 1, & \text{with probability 0.5} \\ 2, & \text{with probability 0.5} \end{cases}$$

The CRI ends at the beginning of slot t , if and only if $x_{t-1} = E$ and $x_{t-2} = E$ or S , and there has been no empty or successful slot followed by an empty slot pattern previously occurred during the CRI. That is, the CRI ends the first time after its beginning, that a noncollision slot is followed by an empty slot.

We note that the operations of the MTCWA within a CRI can be depicted by a two-cell stack, where at each time t , cell 1 contains the transmitting users (those with $r_t = 1$), and cell 2 contains the withholding users (those with $r_t = 2$). As dictated by the algorithm rules following a noncollision slot (E or S slot), all the nontransmitted packets

in the stack move to the transmission cell 1, and cell 2 becomes empty. Thus, given some CRI whose length is more than one slot (that is, a nonempty examined interval), the first time that a noncollision slot is followed by an empty slot all users come to the knowledge that the stack is empty; therefore, that the CRI has ended.

As mentioned earlier the MTCWA is a modification of the algorithm in [10], where the latter operates with binary, collision versus noncollision, feedback. The difference between the two algorithms lies in steps 2, 3.2.(c), and 3.2.(d) in the description of the MTCWA, and in the identification of the slot when some CRI, whose length is more than one slot ends. Indeed, the algorithm in [10] was designed for systems where a success feedback implies single transmission. In the unmodified algorithm a CRI whose first slot is success lasts one slot, and if during some CRI $x_t = \text{noncollision}$ and $r_t = 1$ occurs, then the single transmission departs the system at slot t . In addition in the algorithm in [10], a CRI whose first slot is a collision slot ends with two consecutive noncollision slots, such that the last slot in the pair is not necessarily empty. As compared to the algorithm in [10] the MTCWA wastes occasionally an additional empty slot at the end of each CRI. This "wasted slot" is an an empty slot appended to the end of CRIs so that users can identify the end of the CRI. It will be demonstrated in Section 3.2 that the beneficial effects of capture more than compensate for these wasted slots.

We point out that in environments with no capture and sure success when single transmissions occur, the algorithm in [10] attains throughput 0.429, which is the same with that attained by the Capetanakis dynamic algorithm [4]. As compared to the latter, the algorithm in [10] has better delay characteristics, and superior performance in the presence of feedback errors. Furthermore, this algorithm can be easily modified to

operate in limited feedback sensing environments, and in contrast to Gallager's algorithm [5], it operates in environments where the Poisson user model is not valid.

3.2 Algorithm Analysis

Let P_k denote the probability of capture, given k simultaneous transmissions, where the set $\{P_k\}_{k \geq 1}$ is as in Chapter 2. Let us define,

$0 \leq n \leq k; L_{n, k-n}$: The expected number of slots needed by the MTCWA for the successful transmission of k packets, given that n of the k packets have counter values equal to 1 and that the remaining $k-n$ packets have counter values equal to 2.

The algorithm rules in Section 3.1 induce the following recursions; where w.p. means with probability.

$$\begin{aligned}
 L_{0,0} &= 1, L_{0,k} = 1 + L_{k,0}; k \geq 1 \\
 k \geq 1; L_{1,k-1} &= \begin{cases} 1 + L_{k-1,0}; \text{w.p. } P_1 \\ 1 + L_{1,k-1}; \text{w.p. } \frac{1}{2}(1-P_1) \\ 1 + L_{0,k}; \text{w.p. } \frac{1}{2}(1-P_1) \end{cases} \quad (3.1) \\
 2 \leq n \leq k; L_{n,k-n} &= \begin{cases} 1 + L_{k-1,0}; \text{w.p. } P_n \\ 1 + L_{i,k-i}; \text{w.p. } \binom{n}{i} 2^{-n}(1-P_n), 0 \leq i \leq n \end{cases}
 \end{aligned}$$

We are concerned with the throughput and delay analysis of the algorithm in the presence of the limit Poisson user model. As discussed in [3] the latter user model

provides a performance lower bound for the MTCWA within the class of independent and identical users whose packet-generating process is memoryless. Let λ denote the intensity of the Poisson traffic process. Given the window Δ of the algorithm, let $E\{l|\Delta, d\}$ denote the expected length of a CRI, given that it starts with an examined interval of length Δ and with a lag d . Then for $\{L_{k,0}\}$ as in (3.1), we obtain:

$$E\{l|\Delta, d\} = \sum_{k=0}^{\infty} L_{k,0} e^{-\lambda\Delta} \frac{(\lambda\Delta)^k}{k!} \quad (3.2)$$

Let the system start operating at time zero, and let us consider the sequence in time of lags that are induced by the algorithm. Let C_i denote the length of the i -th lag, where $i \geq 1$. The first lag corresponds to the empty slot zero; thus, $C_1 = 1$. In addition the sequence $C_i; i \geq 1$ is a Markov chain whose state space is at most countable. Let D_n denote the delay experienced by the n -th successful transmission. Let the sequence $T_i; i \geq 1$ be defined as follows: Each T_i corresponds to the beginning of some slot, and $T_1 = 1$. Also, each T_i corresponds to the ending point of a length-one lag. T_{i+1} is then the ending point of the first after T_i unity length lag. Let $R_i; i \geq 1$ denote the number of successfully transmitted packets in the interval $(T_i, T_{i+1}]$. Let $Q_i = R_{i+1} - R_i$. The sequence $Q_i; i \geq 1$ is a sequence of i.i.d. random variables; thus, $R_i (i \geq 1)$, is a renewal process. In addition the delay process $D_n (n \geq 1)$ induced by the algorithm is regenerative with respect to the process $R_i; i \geq 1$ and the distribution of Q_i is nonperiodic since $P(Q_i = 1) > 0$.

Let us define,

$$Z = E\{Q_1\}, W = E\left\{\sum_{i=1}^{Q_1} D_i\right\} \quad (3.3)$$

From the regenerative arguments in [6], it follows that the expected steady-state delay, D per successfully transmitted packet is given by the following expression:

$$D = WZ^{-1} \quad (3.4)$$

The effective computation of D relies on the successful derivation of upper and lower bounds on the quantities W and Z . Those bounds are found via the utilization of the methodology in [5], in conjunction with the quantities in Appendix I. The bounds on W and Z can be found only if:

$$\Delta > E\{I|\Delta, d\}, \quad (3.5)$$

where $E\{I|\Delta, d\}$ is as in (3.2), and where (3.5) determines the stability region of the algorithm. Note the delay per packet transmitted at time t is on the order of the lag at t ; recall the transitions in time of the lag define a Markov chain. Satisfaction of (3.5) guarantees a negative drift for this chain, which from Pakes Lemma [19] insures the ergodicity of the chain.

For various values of the probabilities p and q , which generate the set $\{P_k\}_{k \geq 1}$ of capture probabilities, we computed the optimal window sizes Δ^* as well as lower and upper bounds, λ_l^* and λ_u^* respectively, on the throughput λ^* of the algorithm. We also computed lower and upper bounds, D^l and D^u respectively, on the expected per packet delay D for various Poisson rates λ within the corresponding stability regions of the algorithm. We include the window sizes and the bounds on the throughputs in Table 3.1. In Table 3.2 we include delay bounds.

p	q	Δ	$\lambda_i = \lambda_j$
1	0	3.59	0.3404
0.99	0	3.60	0.3381
0.95	0	3.65	0.3283
0.9	0	3.73	0.3159
0.8	0	3.90	0.2900
0.7	0	4.08	0.2626
0.6	0	4.33	0.2335
0.5	0	4.62	0.2024
1	0.2	3.61	0.3439
0.99	0.2	3.63	0.3464
0.95	0.2	3.69	0.3364
0.9	0.2	3.76	0.3235
0.8	0.2	3.94	0.2968
0.7	0.2	4.14	0.2684
0.6	0.2	4.38	0.2384
0.5	0.2	4.70	0.2064
1	0.4	3.76	0.3767
0.99	0.4	3.77	0.3740
0.95	0.4	3.84	0.3628
0.9	0.4	3.93	0.3484
0.8	0.4	4.12	0.3187
0.7	0.4	4.36	0.2873
0.6	0.4	4.64	0.2542
0.5	0.4	5.02	0.2192
1	0.6	4.12	0.4311
0.99	0.6	4.5	0.4277
0.95	0.6	4.23	0.4141
0.9	0.6	4.35	0.3968
0.8	0.6	4.62	0.3609
0.7	0.6	4.93	0.3234
0.6	0.6	5.31	0.2842
0.5	0.6	5.82	0.2431
1	0.8	5.14	0.5364
0.99	0.8	5.17	0.5319
0.95	0.8	5.33	0.5133
0.9	0.8	5.53	0.4898
0.8	0.8	6.01	0.4415
0.7	0.8	6.59	0.3916
0.6	0.8	7.31	0.3401
0.5	0.8	8.24	0.2871
1	0.9	6.60	0.6388
0.99	0.9	6.66	0.6330
0.95	0.9	6.92	0.6096
0.9	0.9	7.28	0.5799
0.8	0.9	8.12	0.5193
0.7	0.9	9.19	0.4572
0.6	0.9	10.59	0.3937
0.5	0.9	12.47	0.3291

Table 3.1

Optimal Window Sizes and Throughputs for the MTCWA

	λ	D^*	D^*
$p=1.0$	0.050	1.3484	1.3704
	0.100	2.3103	2.3638
	0.150	2.9544	3.1088
	0.200	4.0071	4.3205
	0.250	6.1368	6.7504
$q=0$	0.300	13.3007	14.8254
	0.340	1128.5511	1303.8993
	0.050	1.7980	1.8199
	0.100	2.1735	2.2354
$p=1.0$	0.150	2.6488	2.8091
	0.200	3.3480	3.6667
	0.250	4.5412	5.1125
	0.300	7.2210	8.2560
	0.350	19.5308	22.5181
$q=0.4$	0.370	74.6573	86.7865
	0.050	1.7653	1.7686
	0.100	2.0472	2.0580
	0.150	2.3460	2.3728
$p=1.0$	0.200	2.6564	2.7122
	0.250	2.9848	3.0962
	0.300	3.3392	3.5613
	0.350	3.7235	4.1338
	0.400	4.1653	4.8773
$q=0.9$	0.450	4.7195	5.9150
	0.500	5.5399	7.5310
	0.550	7.1278	10.6369
	0.600	12.5655	21.0146
	0.638	467.0843	856.2050
$p=0.8$	0.050	2.2638	2.3986
	0.100	2.8396	3.1943
	0.150	3.8213	4.4818
	0.200	5.8827	7.0503
	0.250	12.9436	15.5750
$q=0.0$	0.290	23541.1719	37671.2773
	0.050	2.2023	2.3529
	0.100	2.6304	3.0088
	0.150	3.3118	3.9728
$p=0.8$	0.200	4.5661	5.6930
	0.250	7.6565	9.5865
	0.300	26.5528	33.0198
	0.318	690.8590	873.2444

λ	D^*	D^*	
0.050	2.2769	2.3020	$p=0.8$
0.100	2.7049	2.7731	
0.150	3.1467	3.2753	
0.200	3.6102	3.8893	
0.250	4.1002	4.6461	
0.300	4.6402	5.6259	$q=0.9$
0.350	5.3091	7.0029	
0.400	6.3227	9.2068	
0.450	8.5711	13.9079	
0.500	21.9793	41.0494	
0.519	995.8154	1953.5874	
0.050	3.6923	4.1729	$p=0.5$
0.100	5.2091	6.3705	
0.150	9.8708	12.3195	
0.200	199.2676	250.6904	
0.202	1134.8308	1449.7117	
0.050	3.5400	4.0669	$p=0.5$
0.100	4.6481	5.8919	
0.150	7.6034	9.9150	
0.200	25.6669	32.3335	
0.219	1992.9868	2624.6123	
0.050	4.0370	4.1415	$p=0.5$
0.100	5.1029	5.4230	
0.150	6.2236	7.1026	
0.200	7.5121	9.4523	
0.250	9.5808	13.6043	
0.300	17.6525	29.0654	$q=0.9$
0.329	3261.8750	6069.5474	

Table 3.2

Delay Bounds for the MTCWA

In both Tables we include the $p=1$ and $q=0$ case, which represents sure success in the presence of single transmissions and lack of capture in the presence of multiple transmissions. For this case we note the inferior performance of the MTCWA as compared to the unmodified version of the algorithm in [10]. In Figure 3.1 we plot the upper bound on the throughput against q for various values of the probability p . In Figure 3.2 we plot the upper bound on the throughput against $p > 1/2$ for various values of the probability q . In Figure 3.3 we plot the delay upper bound D^u against the Poisson traffic intensity λ for various values of the probabilities p and q .

From Table 3.1 we observe that in the absence of capture and in the presence of sure success of a single transmission, the MTCWA attains throughput 0.3404; the throughput of the unmodified algorithm is 0.429. This loss in throughput is overcompensated for large enough values of the probabilities p and q . As the latter probabilities increase, the throughput of the MTCWA increases monotonically (see Table 3.1, and Figures 3.1 and 3.2) remaining strictly less than one. As observed from Table 3.1, the optimal window sizes increase with increasing q . This occurs since increasing the probability of capture makes it more efficient to have multiple packets per window, thereby, lessening the probability of empty transmission slots. From Table 3.2 and Figure 3.3 we observe that when the value of the probability q is large, then the expected per packet delays for small Poisson intensities increase as compared to those corresponding to smaller q values. This is expected since as q approaches the value 1 the MTCWA basically operates as the TDMA algorithm, which notoriously induces high delays in the presence of low traffic rates.

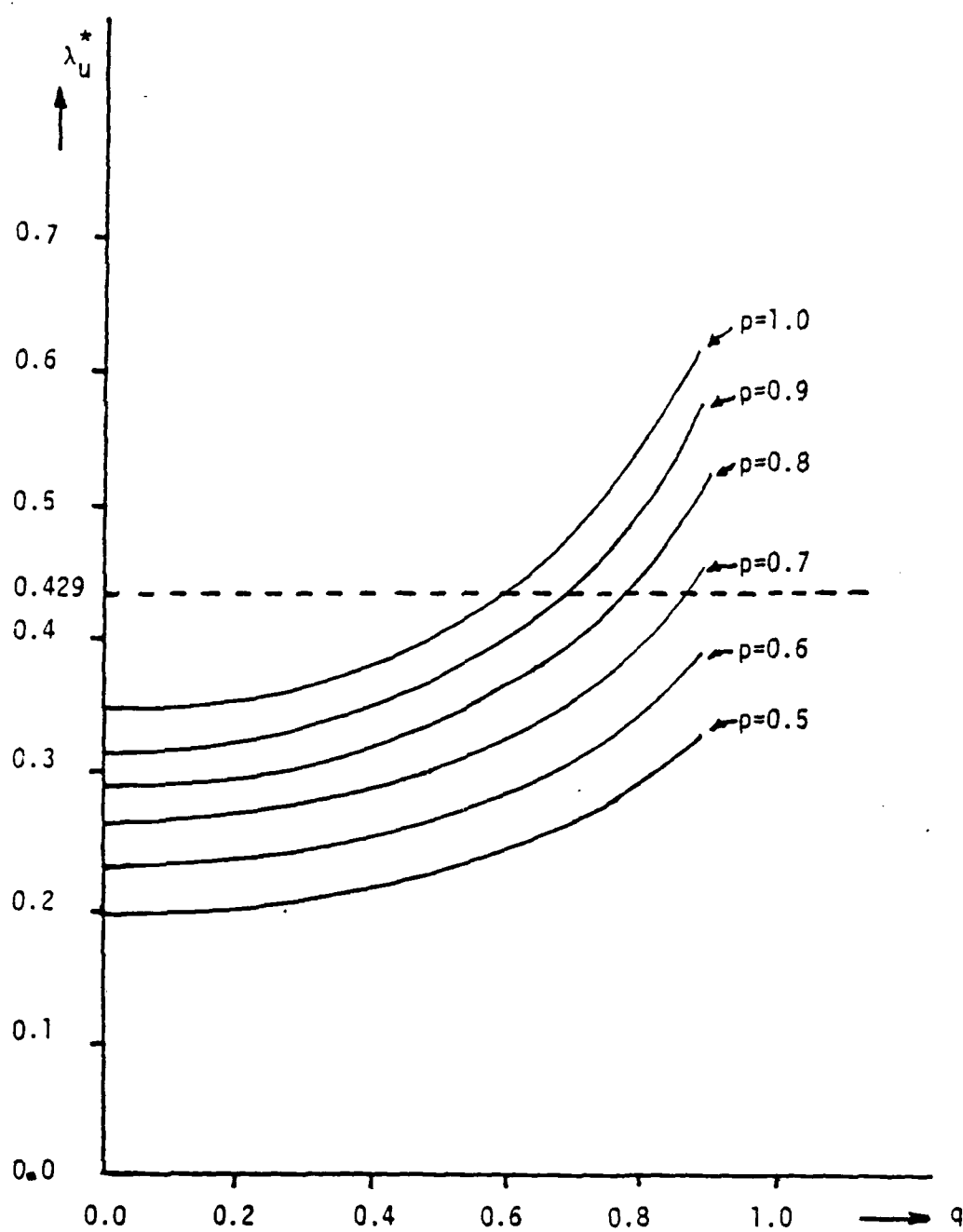


Figure 3.1

The MTCWA Throughput Against q

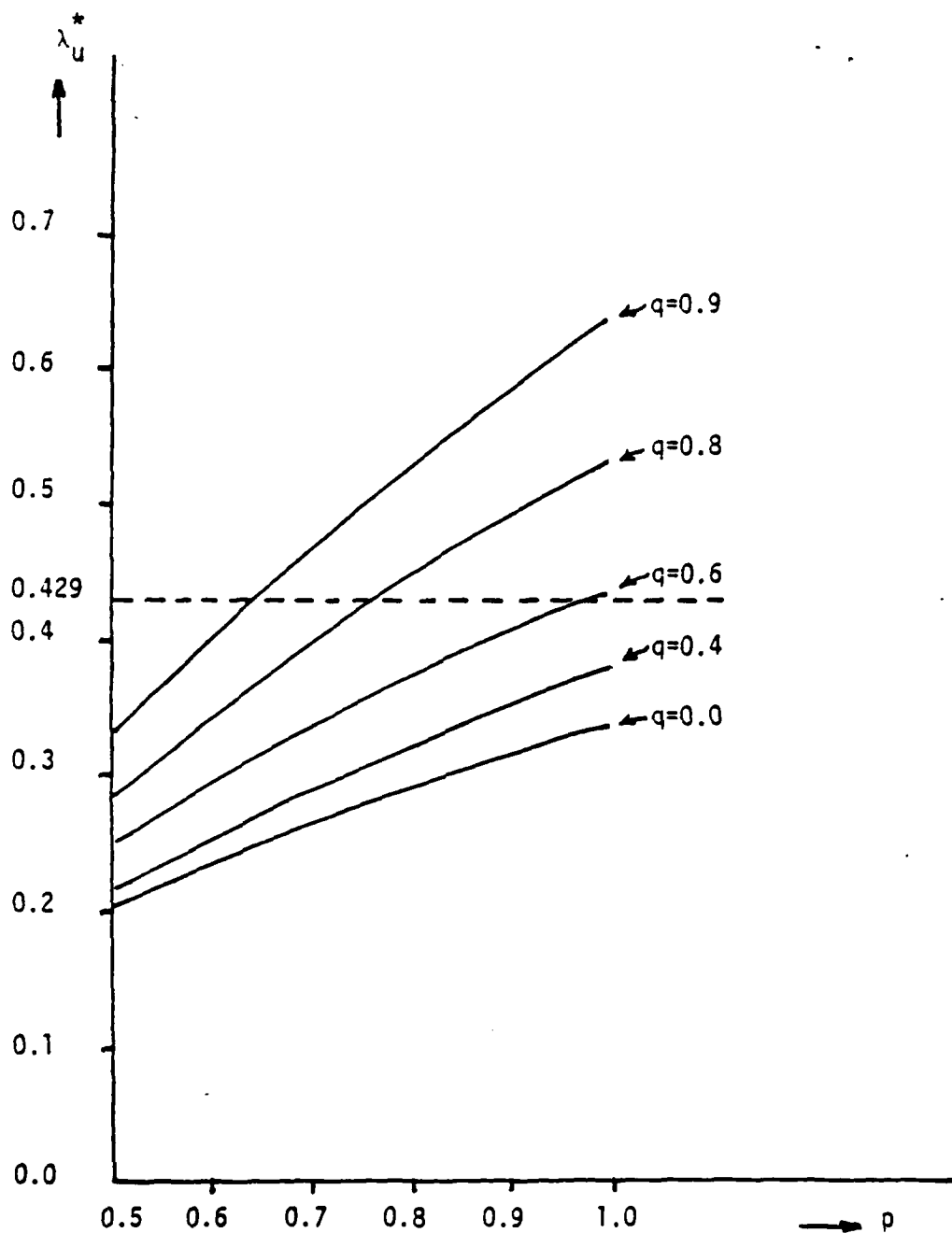


Figure 3.2

The MTCWA Throughput Against p

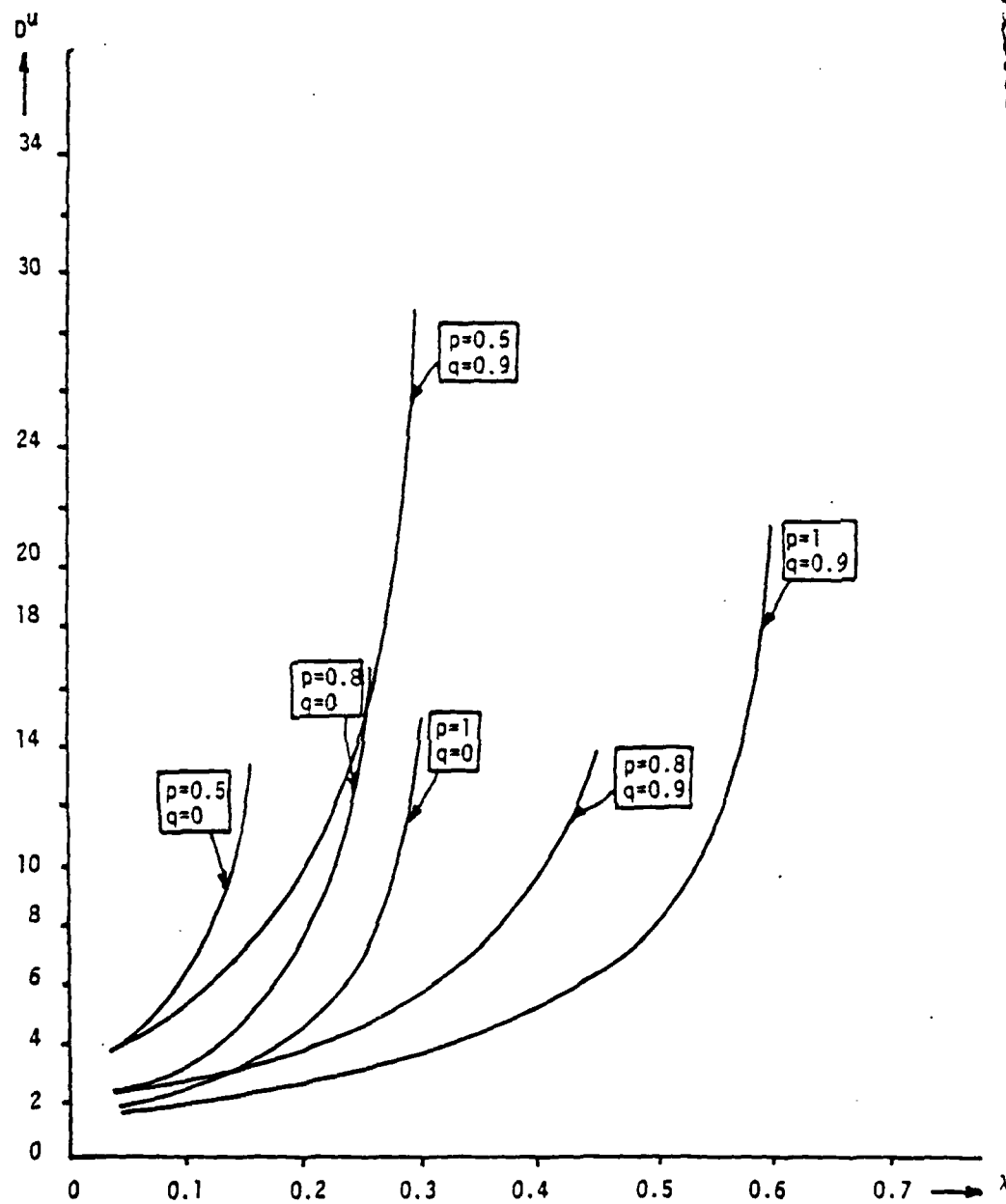


Figure 3.3

Expected per Packet Delays for the MTCWA

We note that the MTCWA maintains many of the advantageous properties of the original algorithm. In particular it can be easily modified to operate in limited-sensing environments, it is highly robust in the presence of feedback errors, and it operates in environments where the Poisson user model is not valid.

3.3 The Capetanakis Algorithm in the Capture Environment

Let us consider the system model in Chapter 1, where the feedback broadcast is ternary, and is augmented by the identification of the successful packet. In this environment a S feedback may correspond to a capture; thus, 1 user is successful and the other users transmitting in that slot must retransmit their packets. The dynamic algorithm of Capetanakis [4], therefore, leads to packet losses unless appropriately modified. The modification is needed for the distinction by all users between success and single transmission versus capture and multiple transmissions. Two reasonable possibilities are the following: (1) After each slot with feedback S, instruct all the users who did not transmit within it to withhold, and the users who might have transmitted within it and were not captured to retransmit. Continue this process until the first non-S slot appears. Otherwise, the algorithm operates as in [4]. (2) After each slot with feedback S, all the users who might have transmitted within it and were not captured continue to transmit until either an E or C feedback occurs. If C occurs the algorithm operates as in [4]. If E is observed all users in the original interval have then been successfully received and a new interval is selected.

Among the above two modifications of the Capetanakis dynamic algorithm, the second is generally more efficient. A form of this modification that can be easily implemented and analyzed is the algorithm in Section 3.2. In this manner the two-cell algorithm in [10] may be seen itself to be a modification of the Capetanakis dynamic algorithm [4].

The first modification described is a more straightforward change to the algorithm in [4] and will be referred to as modification-1 of the Capetanakis dynamic algorithm. The first modification induces the following recursions, where L_k denotes the expected number of slots needed for the resolution of a multiplicity- k collision:

$$\begin{aligned}
 L_0 &= 1 \\
 L_1 &= \begin{cases} 2 & ; \text{w.p. } P_1 \\ 2 + L_1 & ; \text{w.p. } (1 - P_1) \end{cases} \\
 k \geq 2; \quad L_k &= \begin{cases} 2 + L_{k-1} & ; \text{w.p. } P_k \\ 1 + L_i + L_{k-i} & ; \text{w.p. } (1 - P_k) \binom{k}{i} 2^{-k}, 0 \leq i \leq k \end{cases}
 \end{aligned} \tag{3.6}$$

From the recursions in (3.6) and via the same methodology as that used for the analysis of the MTCWA, we computed optimal window sizes and tight throughput and delay bounds. Bounds were computed for the limit Poisson user model and for various values of the probabilities p and q , where $P_k = pq^{k-1}$. We include the optimal window sizes and the throughput bounds in Table 3.3. In Figure 3.4 we plot throughput against q for various values of the probability p for both the MTCWA and modification-1 of the Capetanakis dynamic algorithm. In Figure 3.5 we plot throughput against p for both the above algorithms and various q values. Finally in Figure 3.6, we plot the delays induced

p	q	Δ	$\lambda_i = \lambda_j$
1.0	0	3.82	0.3005
0.99	0	3.85	0.2986
0.95	0	3.95	0.2912
0.9	0	4.08	0.2817
0.8	0	4.40	0.2612
0.7	0	4.81	0.2389
0.6	0	5.36	0.2145
0.5	0	6.13	0.1877
1.0	0.2	3.86	0.3068
0.99	0.2	3.88	0.3049
0.95	0.2	3.98	0.2972
0.9	0.2	4.12	0.2871
0.8	0.2	4.45	0.2657
0.7	0.2	4.87	0.2426
0.6	0.2	5.43	0.2174
0.5	0.2	6.22	0.1898
1.0	0.4	3.98	0.3260
0.99	0.4	4.01	0.3238
0.95	0.4	4.13	0.3150
0.9	0.4	4.28	0.3036
0.8	0.4	4.64	0.2796
0.7	0.4	5.10	0.2539
0.6	0.4	5.70	0.2262
0.5	0.4	6.57	0.1964
1.0	0.6	4.24	0.3584
0.99	0.6	4.28	0.3558
0.95	0.6	4.41	0.3452
0.9	0.6	4.59	0.3317
0.8	0.6	5.03	0.3036
0.7	0.6	5.59	0.2737
0.6	0.6	6.32	0.2420
0.5	0.6	7.37	0.2083
1.0	0.8	4.81	0.4076
0.99	0.8	4.86	0.4045
0.95	0.8	5.05	0.3918
0.9	0.8	5.33	0.3755
0.8	0.8	5.97	0.3416
0.7	0.8	6.80	0.3057
0.6	0.8	7.93	0.2679
0.5	0.8	9.53	0.2282
1.0	0.9	5.48	0.4431
0.99	0.9	5.56	0.4397
0.95	0.9	5.85	0.4258
0.9	0.9	6.17	0.4079
0.8	0.9	7.28	0.3704
0.7	0.9	8.64	0.3306
0.6	0.9	10.56	0.2885
0.5	0.9	13.35	0.2442

Table 3.3
Optimal Window Sizes and Throughputs for Modification-1 of the
Capetanakis Dynamic Algorithm

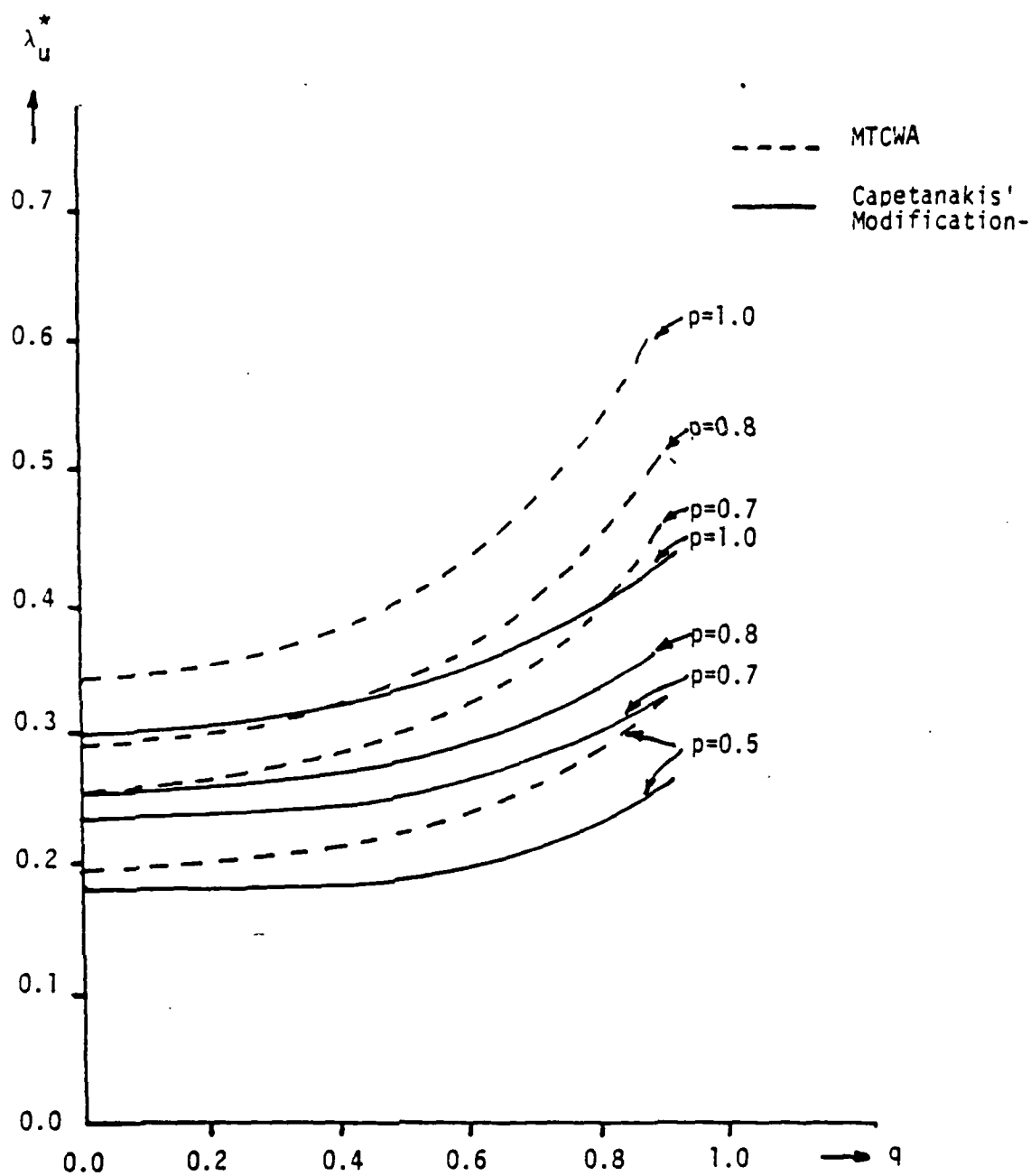


Figure 3.4
Throughputs of the MTCWA and of Modification-1 of
Capetanakis' Dynamic Algorithm Against q

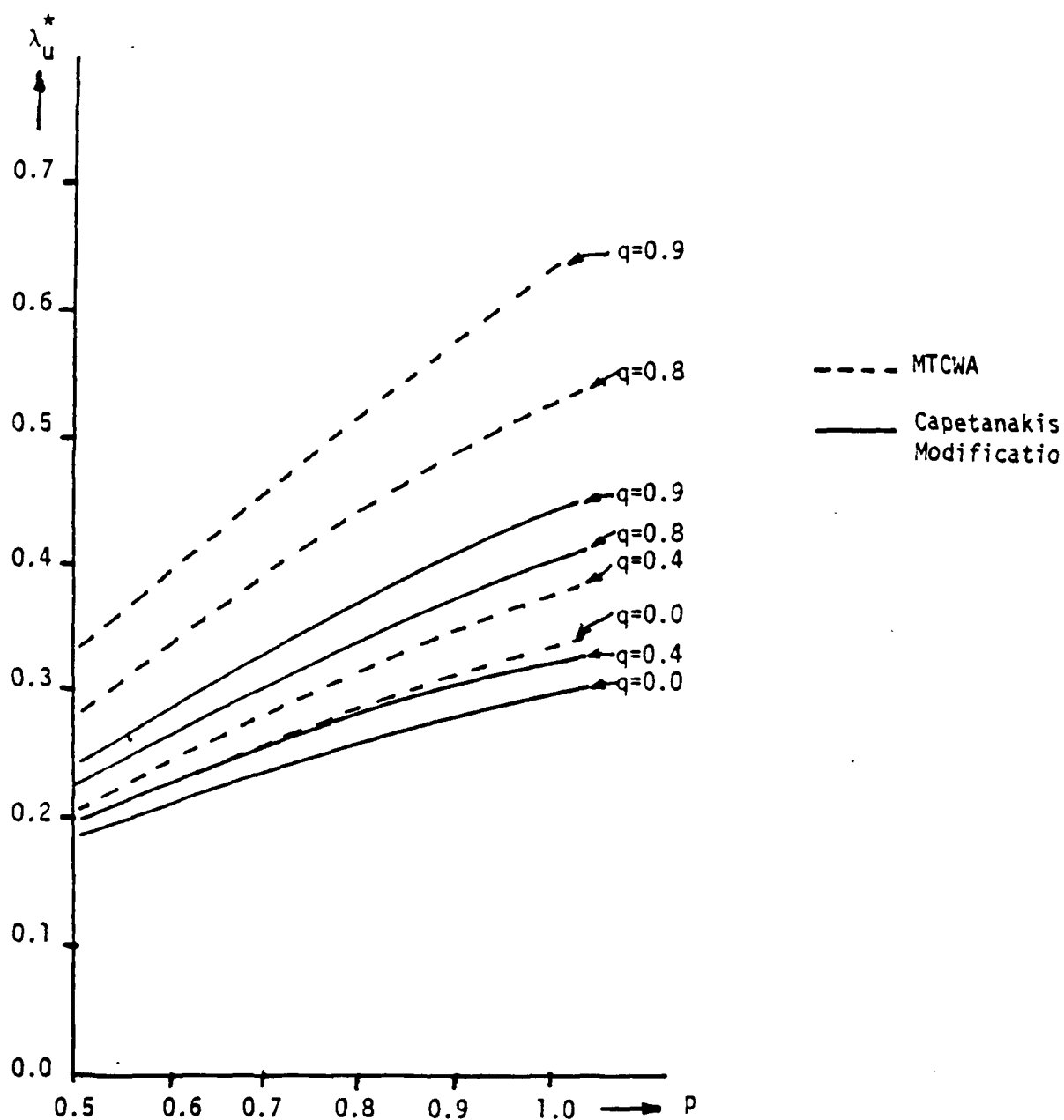


Figure 3.5
Throughputs of the MTCWA and of Modification-1 of
Capetanakis' Dynamic Algorithm Against p

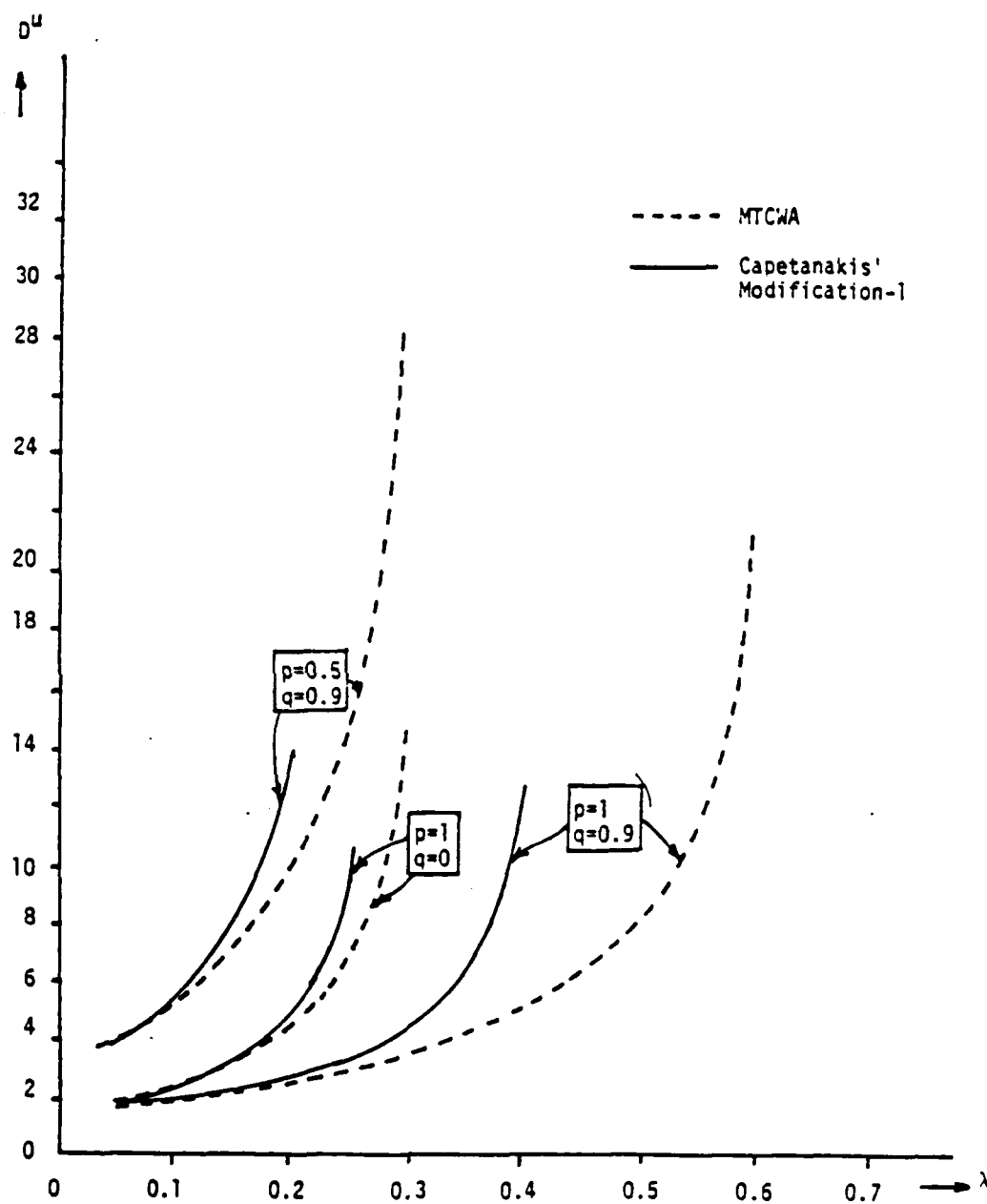


Figure 3.6
Expected per Packet Delays for the MTCWA and for Modification-1
of Capetanakis' Dynamic Algorithm

by the two algorithms as functions of the Poisson traffic intensity λ , and for various p and q values.

Comparing Table 3.3, Figures 3.4 and 3.5 with Table 3.1, Figures 3.1 and 3.2, as well as Figure 3.3 with Figure 3.6, we observe the uniformly worse performance of the Capetanakis modification-1 algorithm compared to the MTCWA. For the same set of capture probabilities the MTCWA provides higher throughputs than does the modified Capetanakis algorithm. Moreover, the MTCWA exhibits uniformly lower delays. Furthermore, as with the dynamic algorithm in [4] modification-1 of the Capetanakis dynamic algorithm in this section is less robust to feedback errors than is the MTCWA.

3.4 Conclusions for the MTCWA

We considered a general communication network with a probabilistic capture mechanism. We assumed ternary feedback augmented by the identification of the successful packet, and a generalized capture model. We proposed and analyzed a stable random-access algorithm (MTCWA), which is a modification of the two-cell algorithm in [10]. The MTCWA can attain quite high throughput and low delays when capture occurs with high probability. In addition it is highly robust in the presence of feedback errors, it can be easily modified to operate in limited sensing environments, and it operates in systems where the Poisson user model is not valid.

CHAPTER 4

An Algorithm for Simple Ternary Feedback

In the previous chapter a high-performance, random-access algorithm was developed, which exploited captures in the presence of multiple transmissions. However, an important assumption in the previous analysis was the existence of an enriched feedback scheme that broadcast to all users the identity of the captured packet. In this chapter an algorithm is proposed for a system with only ternary feedback. This algorithm would be appropriate for networks where capture is an undesired and possibly undetectable feature. Furthermore, only a small-bandwidth, ternary-feedback channel is required to implement the algorithm.

4.1 Description of Algorithm for Simple Ternary Feedback

The proposed algorithm is as in [10] with the following modification:

Each S slot of the original algorithm expands here to $m+1$ slots. In particular if $x_t = S$, then each of the users that transmitted in slot t retransmits in one of the slots $t+1, \dots, t+m$, with probability $1/m$. Just after slot $t+m$ all the packets that transmitted in slot t depart the system, and the original algorithm resumes its operations at time $t+m+1$.

For completeness we describe here the operations of the algorithm. Let t be a time instant that corresponds to the beginning of some slot, and let t_1 be such that $t_1 < t$ and all

the packet arrivals in $(0, t_1]$ have departed at t , and there is no information regarding the arrival interval $(t_1, t]$. Then t is a collision resolution point (CRP), and $t_1 - t$ is the lag at t . The algorithm utilizes a window of length Δ . In slot t the packet arrivals in $(t_1, t_2] \triangleq \min(t_1 + \Delta, t)$ attempt transmission, and the arrival interval $(t_1, t_2]$ is called the "examined interval." The examined interval is resolved when all the arrivals in it have departed the system, and this event is known to all users in the system. Until $(t_1, t_2]$ is resolved no arrivals in $[t_2, \infty)$ are allowed transmission, and the time period required for the resolution of an examined interval is called a collision resolution interval (CRI). The algorithm rules are implemented independently by each user via a counter. The counter value at time t is denoted r_t , where r_t equals either 0, 1, or 2. The counter values are used and updated as follows:

1. A user transmits in some slot t , if and only if $r_t = 1$.
2. At time t when the CRI starts, all users with arrivals in $(t_1, t_2]$ set $r_t = 1$.
 - (a) If $x_t = E$, then the examined interval is resolved at t , and a new CRI starts at $t+1$ with the examined interval $(t_2, \min(t_2 + \Delta, t_2 + 1)]$.
 - (b) If $x_t = S$, then the CRI ends at $t+m$, and a new CRI starts at $t+m+1$ with the examined interval $(t_2, \min(t_2 + \Delta, t_2 + 1 + m)]$. Within the length $m+1$ CRI, a user who transmitted in slot t sets: $r_{t+k} = 0 \quad \forall k : 0 \leq k \leq j, r_{t+j+1} = 1$, with probability $1/m$, where $0 \leq j \leq m-1$. The corresponding packet departs then the system at time $t+j+1$, independently of if it were successfully transmitted or not.

(c) If $x_t = C$, then at time $t+1$, each user who transmitted in t sets:

$$r_{t+1} = \begin{cases} 1 & \text{; with probability } 1/2 \\ 2 & \text{; with probability } 1/2 \end{cases}$$

3. Given a CRI whose first slot is a collision, let $\{t_i\}_{i \geq 1}$ be a sequence of slots in the CRI defined as follows: t_1 is the first slot from the beginning, such that $x_{t_1} = S$. t_{i+1} is the first slot after $t_i + m$, such that $x_{t_{i+1}} = S$. Then,

(a) If $r_{t_i} = 2$, then,

$$r_{t_i + j} = 2, \forall j : 1 \leq j \leq m$$

$$r_{t_i + m + 1} = 1$$

(b) If $r_{t_i} = 1$, then,

$$\left. \begin{array}{l} r_{t_i + k} = 0 ; \forall k : 0 \leq k \leq j \\ r_{t_i + j + 1} = 1 \end{array} \right\} \text{with probability } 1/m$$

and the packet departs then the system at time $t_i + j + 1$.

(c) If $t \neq t_i + j$ for some $1 \leq j \leq m$ and some t_i in $\{t_i\}_{i \geq 1}$, then:

(i) If $r_t = 2$ and $x_t = C$, then set $r_{t+1} = 2$.

(ii) If $r_t = 1$ and $x_t = C$, then set:

$$r_{t+1} = \begin{cases} 1 ; & \text{with probability } 1/2 \\ 2 ; & \text{with probability } 1/2 \end{cases}$$

(iii) If $r_t = 2$ and $x_t = E$, then set $r_{t+1} = 1$.

Let us consider the sequence $\{t_i\}_{i \geq 1}$ of success slots, within a CRI, as defined in Step 3 of the algorithm description. The m slots following each t_i are basically used by the algorithm as nonfeedback or transparent slots. We will call a pattern of $m+1$ consecutive slots which is headed by one of the slots in the sequence $\{t_i\}_{i \geq 1}$ an "extended-S slot." From the description of the algorithm we easily conclude that the following possibilities exist regarding the nature of a CRI: (a) A CRI may consist of a single empty slot. (b) A CRI may consist of a single extended-S slot. (c) A CRI which begins with a collision slot ends with either two consecutive extended-S slots, or an extended-S slot followed by an empty slot, or an empty slot followed by an extended-S slot.

4.2 Analysis of Algorithm

Let $P_k = pq^{k-1}$, $k \geq 1$, denote the probability of capture, given k simultaneous transmissions. We will assume that in the event of capture each of the k packets is captured with probability $1/k$. Let us then define:

$0 \leq n \leq k \leq \infty$; $(n, k-n)$: The event that n packets have counter values equal to 1 and $k-n$ packets have counter values equal to 2, during a CRI which starts with a collision slot.

- $0 \leq n \leq k \leq 0 ; L_{n,k-n}^{(m)} :$ The expected number of slots needed for the resolution of the event $(n,k-n)$, when the algorithm utilizes the integer m in steps 3.a and 3.b of its description.
- $0 \leq n \leq k \leq 0 ; N_{n,k-n}^{(m)} :$ The expected number of lost packets during the resolution of the event $(n,k-n)$, when the algorithm utilizes the integer m in Steps 3.a and 3.b of its operation.
- $k \geq 1 ; X_k^{(m)} :$ The expected number of successfully transmitted packets given k simultaneous transmissions with capture, and given that after the capture event, each of the k packets is transmitted within one of m slots, with probability $1/m$.
- $p_{c,k}^{(m)} :$ Given that the algorithm utilizes the integer m in steps 3.a and 3.b of its operation, given k simultaneous transmissions with capture, given that after the initial capture event each of the k packets is transmitted within one of the m slots with probability $1/m$, the probability that in any one of the m slots a packet is captured and this packet is different than that captured at the initial capture event.
- $0 \leq k ; L_k^{(m)} :$ The expected length of a CRI that starts with k simultaneous transmissions, when the algorithm utilizes the integer m in Steps 3.a and 3.b of its operation.
- $0 \leq k ; N_k^{(m)} :$ The expected number of lost packets throughout the length of a CRI which starts with k simultaneous transmissions, when the

algorithm utilizes the integer m in Steps 3.a and 3.b of its operation.

The algorithm rules induce the following recursions, where w.p. means with probability.

$$L_{0,0} = 1, \quad L_{0,k} = 1 + L_{k,0}; k \geq 1$$

$$\left. \begin{matrix} k \geq n \geq 1 \\ k \geq 1 \end{matrix} \right\}; L_{n,k-n}^{(m)} = \begin{cases} m+1 + L_{k-n,0}^{(m)}; \text{w.p. } P_n \\ 1 + L_{i,k-i}^{(m)}; \text{w.p. } \left[\frac{n}{i} \right] 2^{-n} (1-P_n), 0 \leq i \leq n \end{cases} \quad (4.1)$$

$$N_{0,0}^{(m)} = 0, \quad N_{0,k}^{(m)} = N_{k,0}^{(m)}; k \geq 1$$

$$\left. \begin{matrix} k \geq n \geq 1 \\ k \geq 1 \end{matrix} \right\}; N_{n,k-n}^{(m)} = \begin{cases} n - X_n^{(m)} + N_{k-n,0}^{(m)}; \text{w.p. } P_n \\ N_{i,k-i}^{(m)}; \text{w.p. } \left[\frac{n}{i} \right] 2^{-n} (1-P_n), 0 \leq i \leq n \end{cases} \quad (4.2)$$

$$\left. \begin{matrix} k \geq 2 \\ m \geq 2 \end{matrix} \right\}; p_{c,k}^{(m)} \triangleq \frac{m-1}{m} \sum_{i=1}^{k-1} \left[\frac{k-1}{i} \right] \frac{1}{m^i} \left[\frac{m-1}{m} \right]^{k-1-i} P_i + \frac{1}{m} \sum_{i=1}^{k-1} \left[\frac{k-1}{i} \right] \frac{1}{m^i} \left[\frac{m-1}{m} \right]^{k-1-i} P_{i+1} \frac{i}{i+1} \quad (4.3)$$

$$X_k^{(0)} = 1; k \geq 1, \quad X_1^{(m)} = 1; m \geq 0$$

$$X_k^{(1)} = 1 + \frac{k-1}{k} P_k; k \geq 1 \quad (4.4)$$

$$\left. \begin{matrix} m \geq 2 \\ k \geq 2 \end{matrix} \right\}; X_k^{(m)} = 1 + m p_{c,k}^{(m)}$$

$$N_k^{(m)} = N_{k,0}^{(m)}; k \geq 0 \quad (4.5)$$

$$L_0^{(m)} = 1, \quad L_k^{(m)} = P_k (1+m) + (1-P_k) \left[1 + \sum_{i=0}^k \left[\frac{k}{i} \right] 2^{-k} L_{i,k-i}^{(m)} \right] = L_{k,0}^{(m)} - P_k; k \geq 1 \quad (4.6)$$

Consider the algorithm in Section 4.1, and let the system start operating at time zero. Let us consider the sequence (in time) of lags induced by the algorithm, and let C_i

denote the length of the i -th lag, where $i \geq 1$. The first lag corresponds to the empty slot zero; thus, $C_1 = 1$. In addition the sequence $C_i, i \geq 1$ is a Markov chain whose state space is at most countable. Let D_n denote the delay experienced by the n -th successfully transmitted packet arrival as induced by the algorithm; that is, the time between the arrival instant of the packet and the completion of its successful transmission. Let the sequence $T_i, i \geq 1$, be defined as follows: Each T_i corresponds to the beginning of some slot, and $T_1 = 1$. Each T_i also corresponds to the ending point of a length-one lag, and T_{i+1} is the first after T_i such point. Let $R_i, i \geq 1$, and $F_i, i \geq 1$, denote respectively the number of successfully transmitted and the number of rejected packets in the time interval $(0, T_i]$. Then $Q_i = R_{i+1} - R_i, i \geq 1$, and $G_i = F_{i+1} - F_i, i \geq 1$, denote respectively the number of successfully transmitted and the number of rejected packets in the interval $(T_i, T_{i+1}]$. The sequences $Q_i, i \geq 1$, and $G_i, i \geq 1$, are sequences of i.i.d. random variables when the input traffic process is memoryless (such as Poisson); thus, the sequences $R_i, i \geq 1$, and $F_i, i \geq 1$, are renewal processes. In addition the delay process $D_n, n \geq 1$, induced by the algorithm is regenerative with respect to the process $R_i, i \geq 1$, and the process $Q_i, i \geq 1$ is nonperiodic since $P(Q_i = 1) > 0$.

Let us consider Poisson input traffic with intensity λ . Let ρ and D be, respectively, the fraction of successfully transmitted packets and the expected steady-state delay per successfully transmitted packet. Let us define:

$$Z = E\{Q_1\}, \quad W = E\left\{\sum_{i=1}^{Q_1} D_i\right\}, \quad H = E\{T_2 - T_1\} \quad (4.7)$$

From the regenerative arguments in [3] we conclude:

$$\rho = Z(\lambda H)^{-1} \quad (4.8)$$

$$D = WZ^{-1}$$

The pertinent equations for the evaluation of the quantities Z , W , and H in (4.7), and the subsequent derivation of upper and lower bounds on ρ and D are given in Appendix II.

Given system probabilities p and q , we may select the algorithm parameters m and Δ to fulfill one of the following two objectives.

- (1) Maximize the region of the Poisson traffic intensities, for which the expected delays of the successfully transmitted packets are finite.
- (2) Given some lower bound ρ^* on the fraction ρ of the successfully transmitted packets, maximize the region of the Poisson traffic intensities that satisfy this bound.

Towards the fulfillment of the first objective, we optimized with respect to the window size Δ for various values of the parameter m . We computed the expected delay per successfully transmitted packet, and the fraction of the successfully transmitted packets for various values of the Poisson traffic intensity λ . Towards the fulfillment of the second objective, we optimized with respect to the window size Δ for various values of the parameter m and the bound ρ^* . Delay bounds were also computed on the expected delay per successfully transmitted packet for various values of the Poisson traffic

intensity λ . In all cases we selected $p=1$, since the properties of our algorithm are basically exhibited by the probability (of capture, in substance) q . In the process we computed bounds on the quantities ρ and D for various values of the parameters q , m , Δ , and λ . We include selective such results in Table 4.1, where ρ_l , ρ_u , D_l and D_u denote lower and upper bounds on ρ and lower and upper bounds on D , respectively.

4.3 Numerical Results, Conclusions, and Comparisons

For the sake of comparison the performance of the ternary-feedback algorithm is compared with that of a completely passive system with capture. In the passive system packets are generated according to a Poisson distribution per slot and are transmitted at the beginning of the next slot; thus, average delay for successful transmissions is always 1.5 slots. Retransmissions are not allowed as the passive system operates without feedback. For a given set of p , q , and λ values the fraction of successfully transmitted packets for the passive system may be found in closed form as $\rho = \lambda^{-1} q^{-1} e^{-\lambda} (e^{\lambda q} - 1)$, where ρ is strictly monotonically decreasing with respect to λ .

In Figures 4.1, 4.2, 4.3, 4.4, and 4.5, we plot numerical results when the algorithm is designed to satisfy Objective 1. In Figures 4.6 and 4.7 we plot such results when the algorithm is designed to satisfy Objective 2. In all figures we also plot the corresponding performance of the passive system for comparison.

In our numerical results we selected the zero, one, and two values of the algorithm parameter m . We note that for $m=0$ and $q=0$, the algorithm reduces to that in [10], it is stable, and its throughput equals 0.43. From Figures 4.4 and 4.5 we observe that the $m=0$

	λ	ρ_l	ρ_u	D_l	D_u
$m=0$ $q=0$ $\Delta=2.33$	0.05	1.0	1.0	1.67	1.68
	0.10	1.0	1.0	1.77	1.87
	0.20	1.0	1.0	2.41	2.58
	0.30	1.0	1.0	4.31	4.52
	0.40	1.0	1.0	20.71	23.80
$m=0$ $q=0.5$ $\Delta=2.0$	0.10	0.975	1.000	1.69	1.70
	0.20	0.945	0.991	1.89	1.96
	0.30	0.912	0.956	2.31	2.48
	0.40	0.870	0.912	3.32	3.55
	0.50	0.824	0.866	6.05	6.85
$m=0$ $q=0.75$ $\Delta=2.0$	0.10	0.956	1.000	1.58	1.50
	0.20	0.917	0.962	1.68	1.70
	0.30	0.870	0.914	1.79	1.84
	0.40	0.819	0.860	1.88	2.02
$m=1$ $q=0$ $\Delta=3.0$	0.10	1.00	1.00	2.16	2.27
	0.20	1.00	1.00	4.34	4.87
	0.29	1.00	1.00	45.30	58.20
$m=1$ $q=0.5$ $\Delta=3.0$	0.10	0.973	1.000	1.96	1.98
	0.20	0.942	0.990	2.71	2.85
	0.30	0.848	0.943	5.20	5.75
$m=1$ $q=0.75$ $\Delta=3.5$	0.10	0.967	1.000	1.72	1.74
	0.20	0.912	0.982	1.98	2.09
	0.30	0.854	0.901	2.44	2.58
	0.40	0.793	0.836	3.22	3.42
$m=2$ $q=0$ $\Delta=3.5$	0.05	1.00	1.00	2.06	2.07
	0.10	1.00	1.00	2.97	3.10
	0.20	1.00	1.00	17.91	19.68
$m=2$ $q=0.5$ $\Delta=3.5$	0.05	0.987	1.00	1.79	1.79
	0.10	0.981	1.00	2.39	2.47
	0.20	0.944	0.993	5.19	5.34
	0.25	0.924	0.972	9.93	11.20
$m=2$ $q=0.75$ $\Delta=4.0$	0.10	0.972	1.000	2.08	2.11
	0.20	0.926	0.981	2.88	2.99
	0.30	0.868	0.919	4.32	4.49
	0.40	0.807	0.855	8.57	9.42

Table 4.1

Bounds on Delay and Success Rate

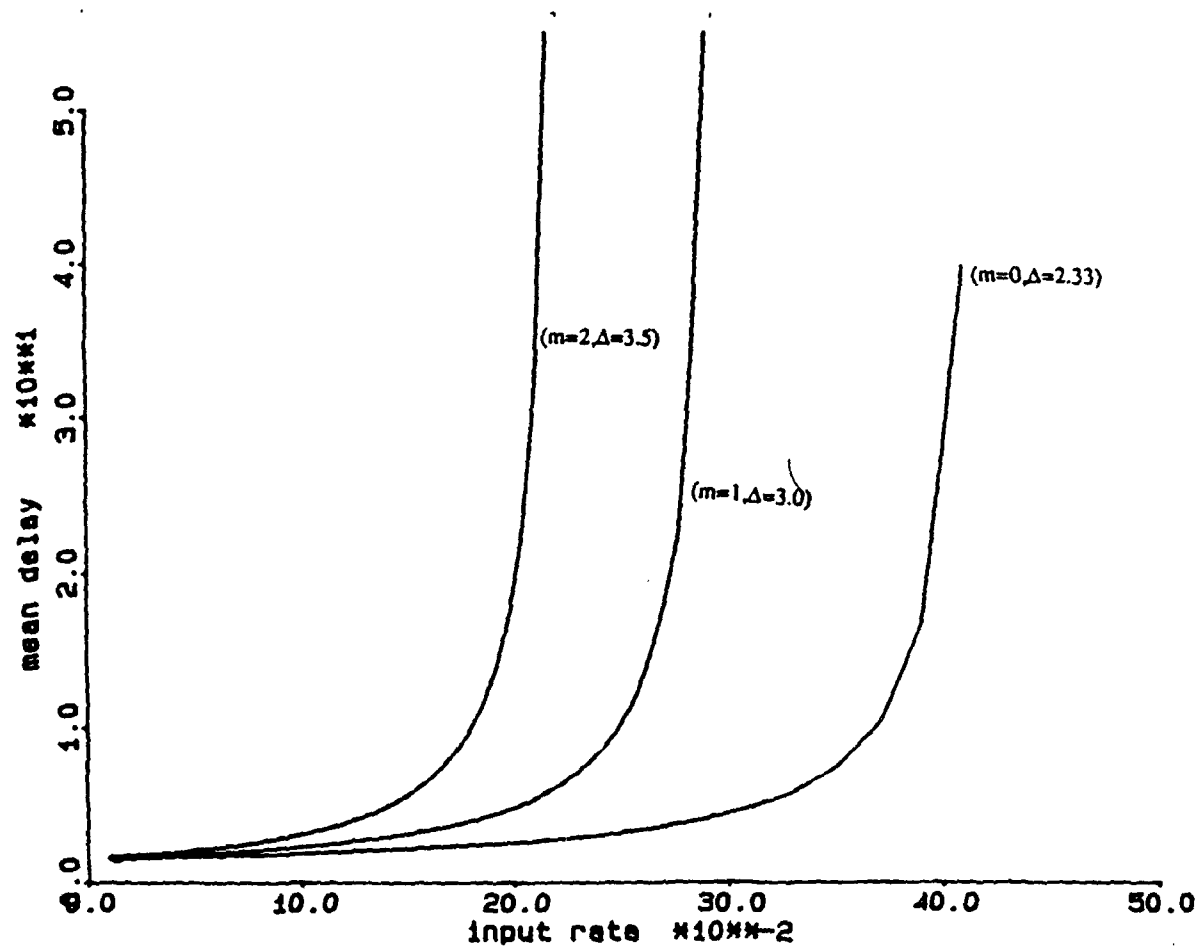


Figure 4.1

Mean Delays for $q=0.0$ (no capture)

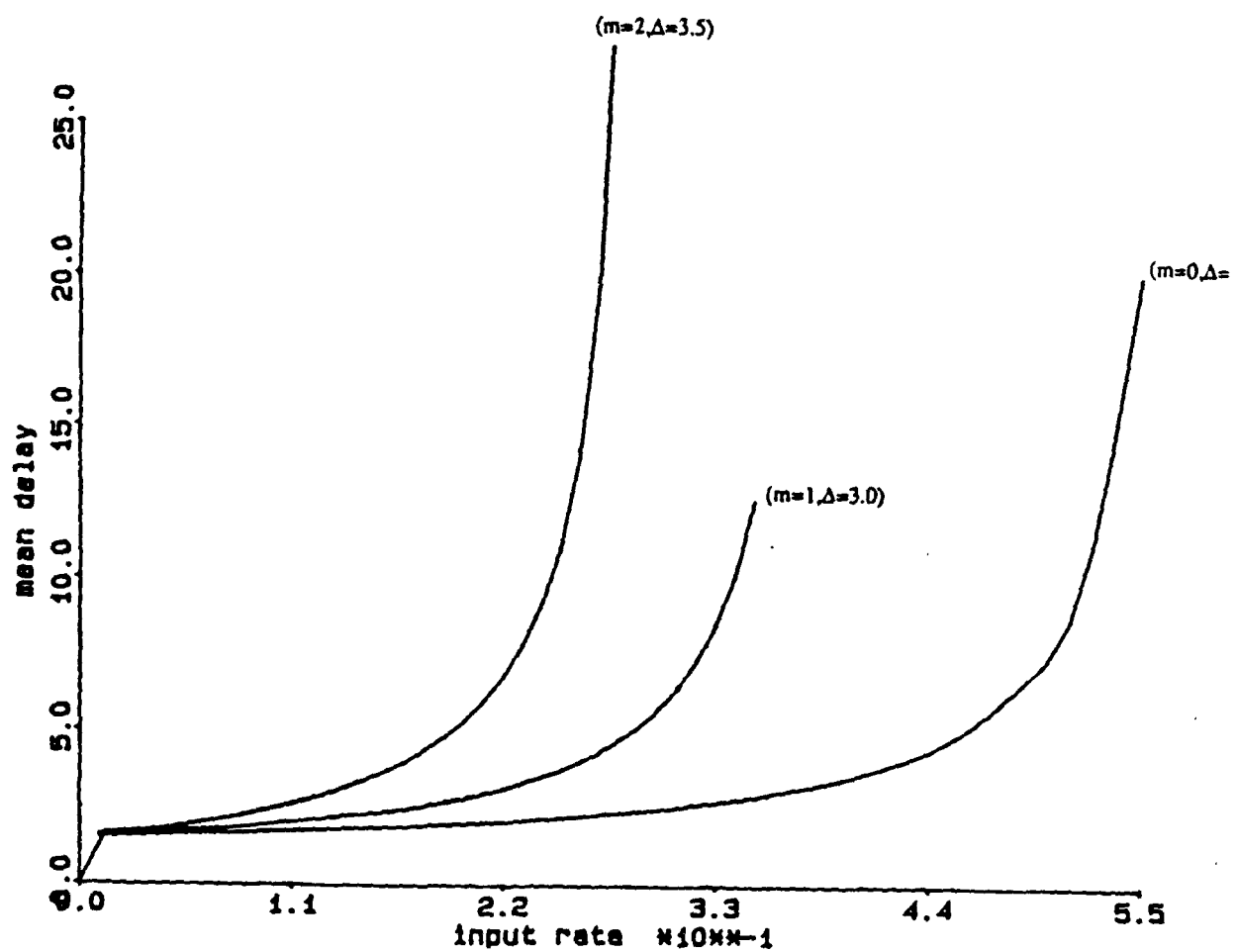


Figure 4.2
Mean Delays for $q = 0.5$

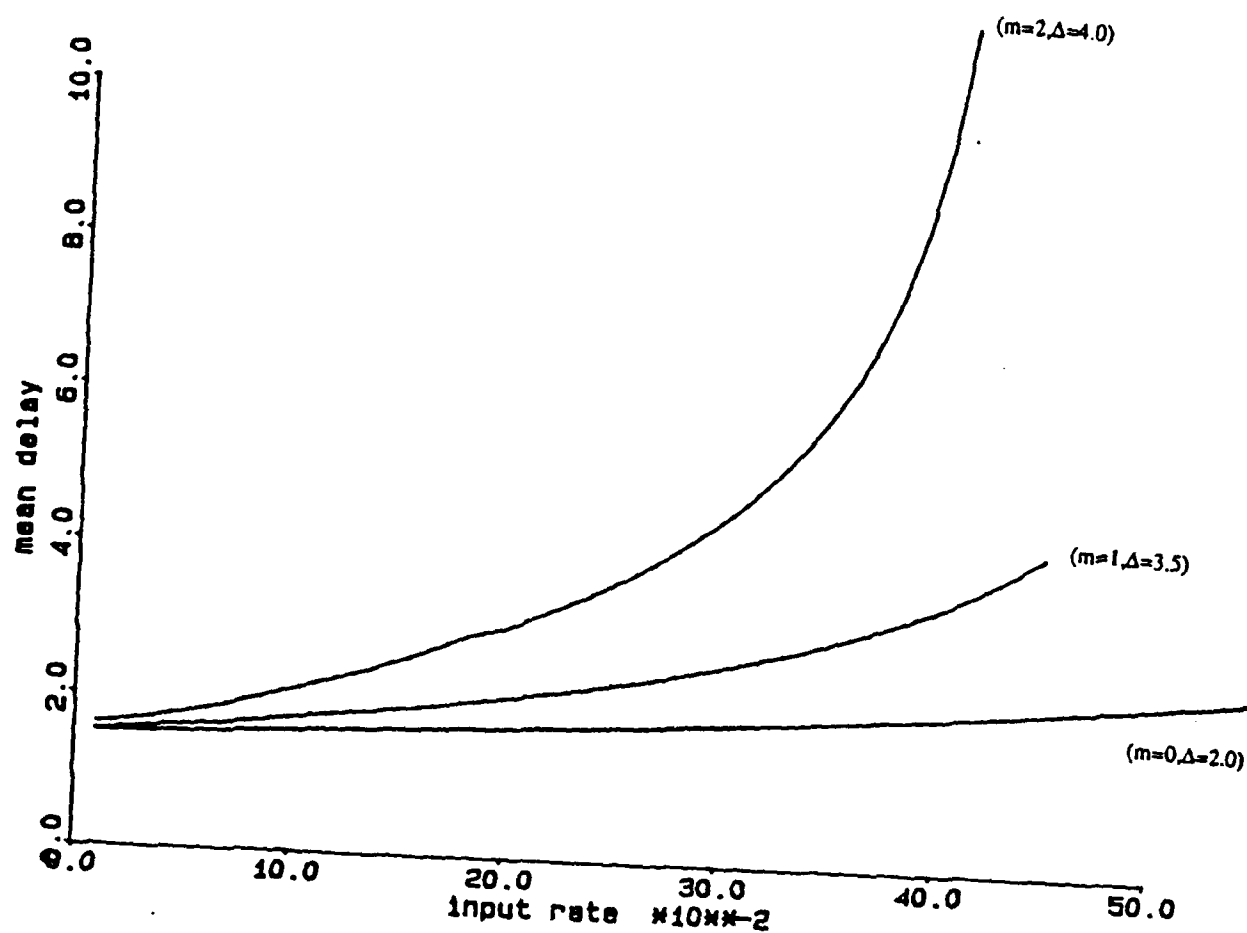


Figure 4.3

Mean Delays for $q = 0.75$

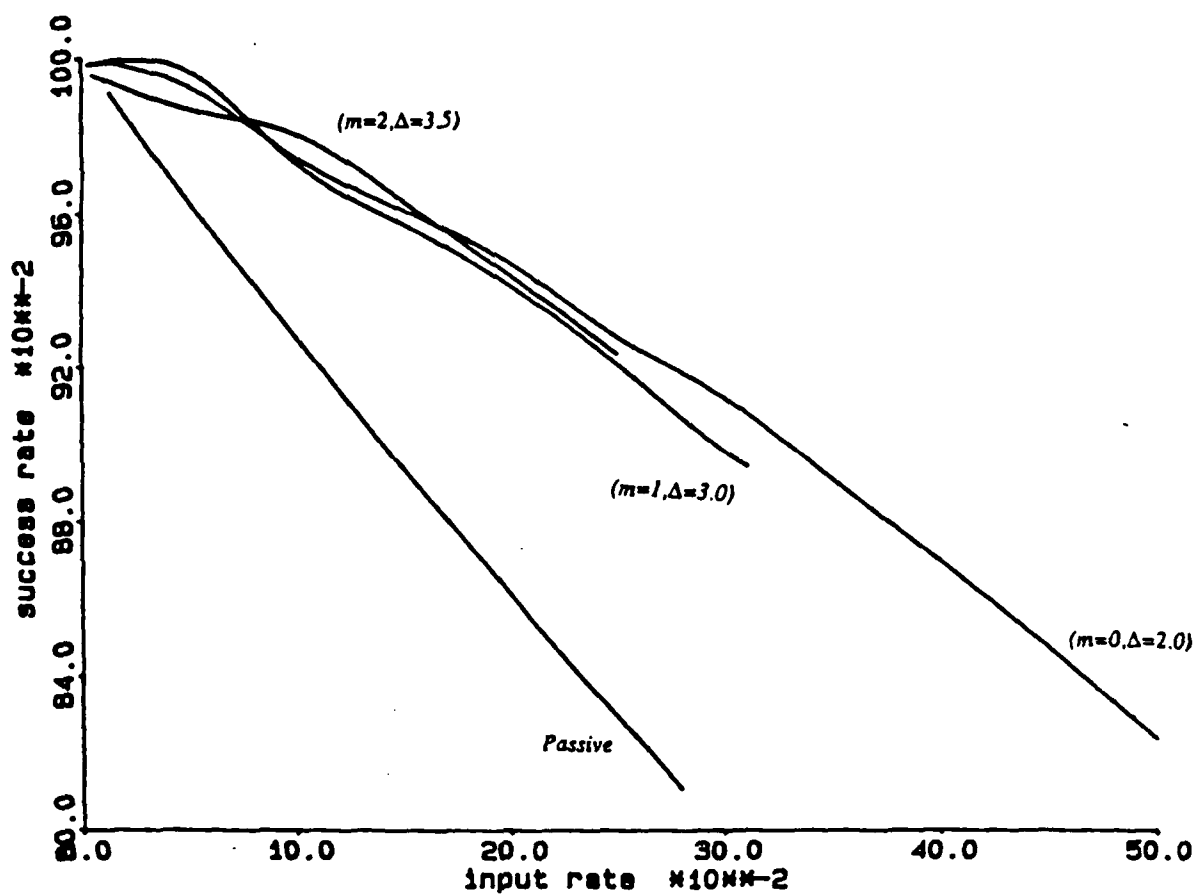


Figure 4.4

Fraction of Traffic Transmitted Successfully (ρ_l) for $q=0.5$

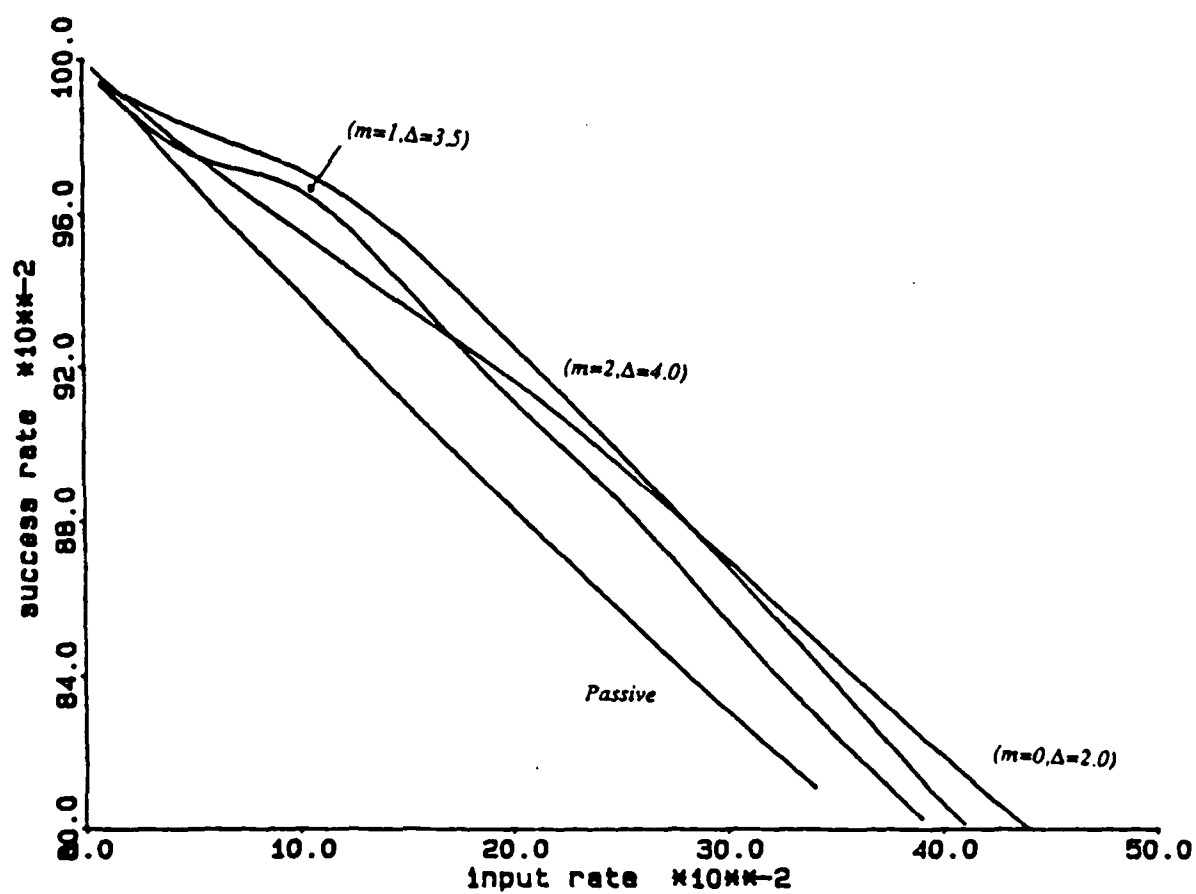


Figure 4.5

Fraction of Traffic Transmitted Successfully (ρ_l) for $q=0.75$

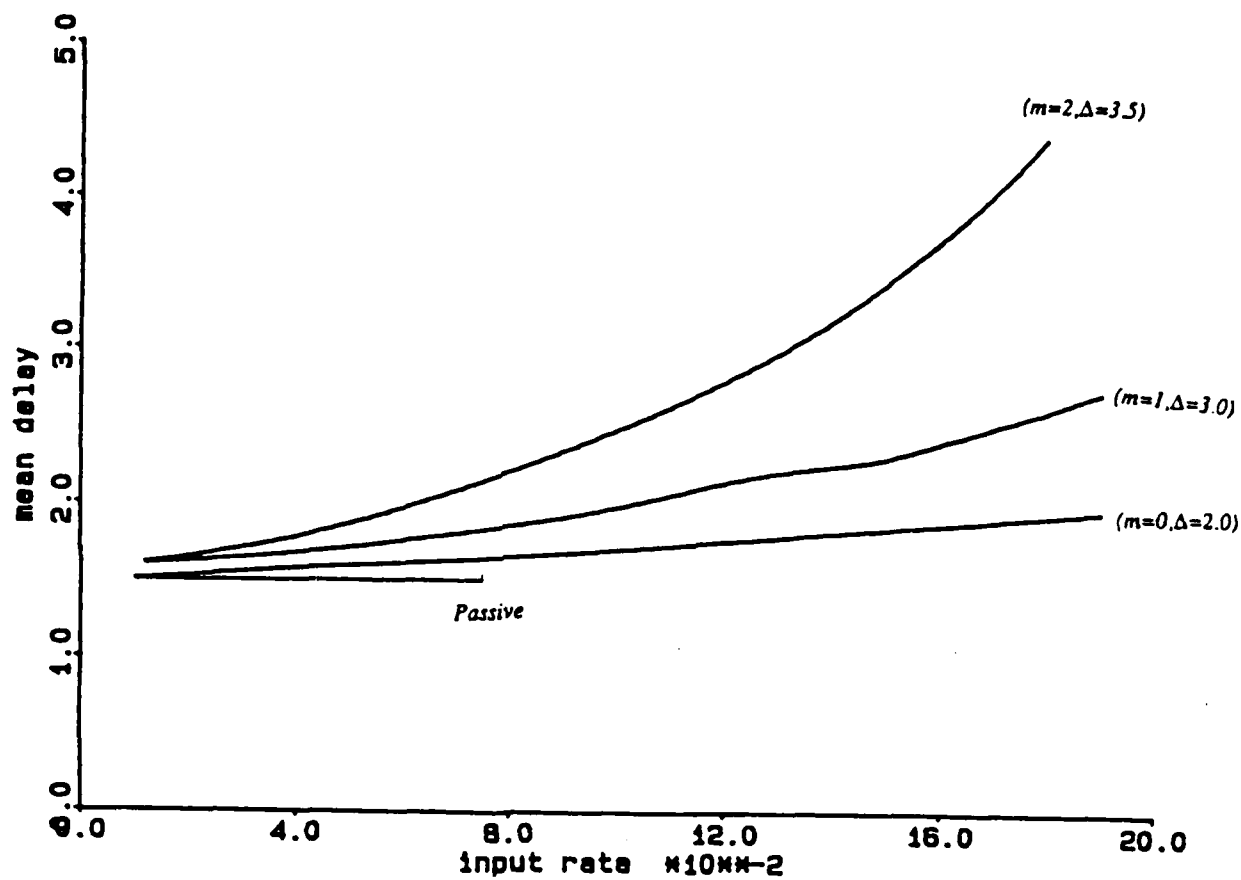


Figure 4.6

Mean Delays for $\rho_l \geq 0.95$, $q=0.5$

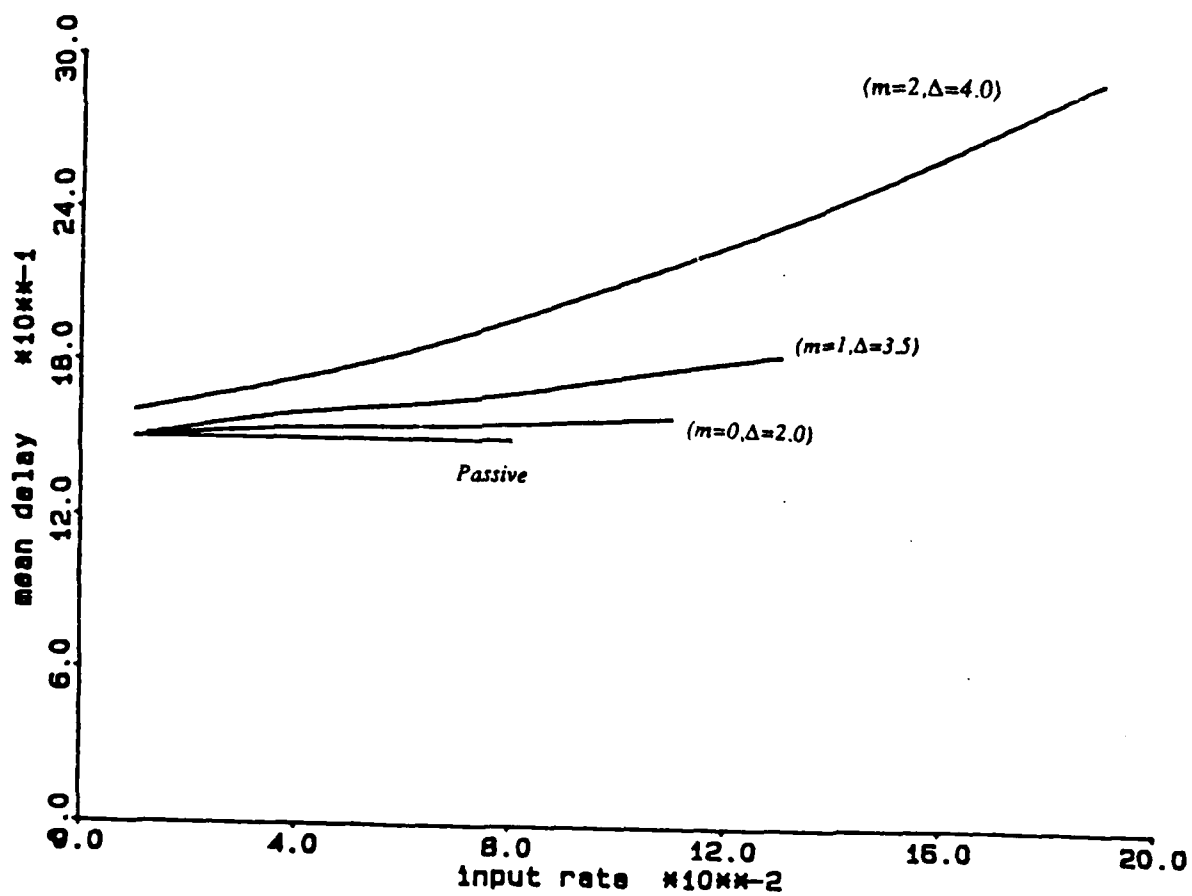


Figure 4.7

Mean Delays for $\rho_l \geq 0.95$, $q = 0.75$

case provides success rates nearly as high or higher than those in the cases of $m=1$ and $m=2$. For example in the low capture probability case of $q=0.5$, the algorithm with $m=0$ maintains a higher success rate than the other algorithms for all Poisson traffic rates above 0.19. Finally it appears that selection of $m=0$ never penalizes the success rate substantially and, in fact, is often the best choice. From Figures 4.4 and 4.5 we also observe that the success rates induced by a passive system are significantly lower than those induced by the algorithm proposed herein. Thus, the simple, ternary-feedback algorithm provides for significant increases in the fraction of successfully transmitted traffic at the expense of slight increases in delay. From all figures we conclude that the $m=0$ selection provides the best delays other than the passive system, which exhibits far worse performance in terms of lost traffic. In addition comparisons between Figures 4.2 and 4.4, and Figures 4.3 and 4.5, lead to the conclusion that the algorithm in this paper, with $m=0$, attains delays close to those induced by a passive system for a significant region of Poisson intensities. Furthermore, it simultaneously outperforms the latter significantly in terms of success rates. The general conclusion drawn from our results is that the basically unmodified two-cell algorithm in [10] performs quite well in the capture environment. One may modify the algorithm as described to increase the success rate; however, a penalty is clearly paid in terms of delays. Inspection of the success rate vs. input rate plots displays immediately that there is no uniform trade-off between delays and success rate.

CHAPTER 5

Conclusions

The preceding chapters have contained sufficient analysis and conclusions based on the results presented in each chapter. However, a few final comments are in order.

5.1 The Two-Cell Algorithm Compared to the Capetanakis Dynamic Algorithm

As described in Chapter 3 the two-cell algorithm is essentially a modification of the Capetanakis dynamic algorithm where the rules of the Capetanakis algorithm have been somewhat simplified. Recall when compared in the no-capture, limit Poisson user model both algorithms attain throughput of 0.429; however, the two-cell algorithm provides uniformly better delays. The results in Chapter 3 demonstrate that the more simple algorithm is more easily and efficiently adapted to the capture environment. It has also been shown [10] that using ternary feedback the two-cell algorithm is far more robust in the presence of feedback errors than is the Capetanakis dynamic algorithm.

5.2 Random-Access Networks with Capture

It is clear from the results presented in Chapter 3 that a random-access algorithm provided with the enriched feedback described can greatly exploit the benefits of capture. The throughput of the enriched feedback algorithm (MTCWA) increased uniformly as

the probability of capture increased.

The average delay per packet tended to decrease also as the capture parameter q was increased from 0 to about 0.9. However, delay performance worsened dramatically as q approached 1. This may be explained as follows. The optimal window size for each set of (p,q) values was chosen so as to maximize the throughput of the algorithm. As the probability of capture approaches one, throughput will be maximized by insuring there is at least one packet in every window. Thus, as $(p,q) \rightarrow (1,1)$ the optimal window size (Δ^*) approaches infinity. Since the average delay per packet is always at least half the window length for these algorithms, it is clear the average delay will also become unbounded. Therefore, in a system with very high probability of capture even for high-multiplicity collisions, a trade-off between throughputs and delays may be necessary. In practice the window size, Δ , would likely be chosen so as to minimize delay at a particular value of the input rate, λ .

5.3 Algorithm Sensitivity to Forward-Channel Errors

Both the algorithms developed in Chapter 3, the MTCWA and the modified Capetanakis Algorithm, were analyzed in the presence of forward-channel errors (that is, $p \neq 1$). In both cases the throughputs were reduced approximately linearly with p . This was to be expected since the throughput is roughly the probability of a correct transmission per slot. A corresponding increase in the delays was also observed; however, both algorithms appeared to be reasonably robust in the presence of channel errors. These observations appear to justify the paradigmatic assumption of an errorless

forward channel. The ternary feedback algorithm of Chapter 4 was, therefore, not analyzed in the presence of such errors.

5.4 Systems Costs to Achieve Capture

Since capture has been shown to enhance greatly the performance of random-access systems, one should logically suspect that it is achieved only at certain costs to the system. Although this paper has not investigated methods for achieving capture, it is at least certain that more complex receivers are necessary at central nodes. Since there are typically many more "user nodes" than central nodes, this may be a very practical improvement. Recall that the transmission-power-randomization techniques require more complex transmitters at all nodes; this implies significant increases in overall system costs. However, other techniques such as time randomization or spread spectrum tend to use channel resources (time and bandwidth, respectively) less efficiently than simpler schemes. It is, therefore, implicit in the analysis presented herein that trade-offs inevitably exist between capture techniques and other system parameters such as complexity and capacity.

References

- [1] Gallager, R.G., and D. Bertsekas, *Data Networks*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1987.
- [2] Tanenbaum, A.S., *Computer Networks*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1981.
- [3] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "On the Relation between the Finite and the Infinite Population Models for a class of RAAs", *IEEE Trans. Comm.*, COM-35, No. 11, 1987.
- [4] J. I. Capetanakis, "Tree Algorithms for Packet Broadcast Channel", *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 505-515, Sept. 1979.
- [5] R. G. Gallager, "Conflict Resolution in Random Access Broadcast Networks", in *Proc. AFOSR Workshop Communication Theory and Applications*, Provincetown, MA, Sept. 1978, pp. 74-76.
- [6] L. Georgiadis, L. Merakos, and P. Papantoni-Kazakos, "A Method for the Delay Analysis of Random Multiple Access Algorithms whose Delay Process is Regenerative", *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 1051-1062, July 1987.

- [7] M. Sidi and I. Cidon, "Splitting Protocols in Presence of Capture", IEEE Trans. Info. Th., Vol. IT-31, pp. 295-301, March 1985.
- [8] D. H. Davis and S. A. Gronemeyer, "Performance of Slotted Aloha Random Access with Delay Capture and Randomized Time of Arrival," IEEE Trans. Commun., vol COM-28, pp. 703-719, May 1980.
- [9] Borchardt, R.L., and T.T. Ha, "Throughput of Slotted ALOHA Networks with Random Multiple Signal Levels," Twenty-Second Annual Conference on Information Sciences and Systems, Princeton, NJ, 1988.
- [10] M. Paterakis and P. Papantoni-Kazakos, "A Simple Window Random Access Algorithm with Advantageous Properties", Report No. UVA/525415/EE88/106, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, July 1987. Also submitted for publication.
- [11] D. C. Cox, "Delay Doppler Characteristics of Multipath Propagation at 910 MHz in a Suburban Mobile Radio Environment," IEEE Trans. Antenna Prop., vol. AP-20, pp. 625-635.
- [12] L. Georgiadis and P. Papantoni-Kazakos, "A 0.487 Throughput Limited Sensing Algorithm", IEEE Trans. on Information Theory, Vol. IT-33, No. 2, March 1987.
- [13] J. L. Metzner, "On Improving Utilization in ALOHA Networks", IEEE Trans. Commun., vol. COM-24, pp. 447-448, Apr. 1976.

- [14] M. Paterakis, L. Georgiadis, and P. Papantoni-Kazakos, "A Full Sensing Window Random-Access Algorithm for Messages with Strict Delay Constraints", Report No. UVA/525415/EE87/103, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, Feb. 1987. Also, submitted for publication.
- [15] B. Ramamurthi, A. M. Saleh, and D. J. Goodman, "Perfect Capture Aloha for Local Radio Communications", IEEE Journal Selec. Ar. Comm., vol. SAC-5, pp. 806-813, June 1987.
- [16] P. Studer and H. Pletscher, "Q-ary Part-and-Try Algorithm for Packet Conflict Resolution", Inst. Telecommun. Swiss Fed. Inst. Technol., Zurich, Switzerland, Oct. 11, 1984.
- [17] D.F. Lyons, and P. Papantoni-Kazakos, "A Window Random Access Algorithm for Environments with Capture", Report No. UVA/525415/EE87/107, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, Sept. 1987. Also to appear in the proceedings of the 1988 International Conference on Communications.
- [18] D.F. Lyons, and P. Papantoni-Kazakos, "A Random Access Algorithm for Environments with Capture and Limited Feedback Capabilities", Report No. UVA/525415/EE87/111, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, March 1988. Also presented at the

Twenty-Second Annual Conference on Information Sciences and Systems,
Princeton, NJ, 1988.

- [19] A.G. Pakes, "Some Conditions for Ergodicity and Recurrence of Markov Chains",
Jou. Operations Research, vol. 17, 1969.

Appendix I

Analysis of Algorithm for Enriched Feedback

This appendix contains the basic quantities necessary to compute the delay and throughput of the algorithm in Chapter 3 for enriched feedback.

Bounds on the $L_{n,k-n}$ Lengths

Given the set $\{P_k\}_{k \geq 1}$ of capture probabilities, let us define,

$$\begin{aligned} A_1^{(0)} &= (3-P_1)(1+P_1)^{-1}, A_1^{(1)} = (1-P_1)(1+P_1)^{-1}, A_1^{(2)} = 2P_1(1+P_1)^{-1} \\ \{A_n^{(0)}\}: A_n^{(0)} &= [1-2^{-n}(1-P_n)]^{-1} \left\{ 1 + 2^{-n}(1-P_n) \sum_{i=1}^{n-1} \binom{n}{i} A_i^{(0)} \right\}, n \geq 2 \end{aligned} \quad (I.1)$$

$$\begin{aligned} \{A_n^{(1)}\}: A_n^{(1)} &= [1-2^{-n}(1-P_n)]^{-1} 2^{-n}(1-P_n) \left\{ 1 + \sum_{i=1}^{n-1} \binom{n}{i} A_i^{(1)} \right\}, n \geq 2 \\ \{A_n^{(2)}\}: A_n^{(2)} &= [1-2^{-n}(1-P_n)]^{-1} \left\{ P_n + 2^{-n}(1-P_n) \sum_{i=1}^{n-1} \binom{n}{i} A_i^{(2)} \right\}, n \geq 2 \end{aligned}$$

Then, from the expressions in (I.1), we easily find by induction:

$$L_{n,k-n} = A_n^{(0)} + A_n^{(1)} L_{k,0} + A_n^{(2)} L_{k-1,0} \quad (I.2)$$

$$L_{k,0} = A_k^{(0)} [1-A_k^{(1)}]^{-1} + A_k^{(2)} [1-A_k^{(1)}]^{-1} L_{k-1,0} \quad (I.3)$$

It can be found by induction, that given n_0 , there exist constants a , b , and c , such that,

$$\begin{aligned}
A_n^{(0)}[1-A_n^{(1)}]^{-1} &\leq a n + b ; \forall n > n_0 \\
A_n^{(1)} &\leq c < 1 ; \forall n > n_0 \\
A_n^{(2)} [1-A_n^{(1)}]^{-1} &\leq 1 ; \forall n > n_0
\end{aligned} \tag{I.4}$$

The bounds in (I.4), in conjunction with (I.3) give:

$$\begin{aligned}
0 \leq L_{k,0} &\leq 2^{-1} a k^2 + (b+2^{-1} a)k - n_0[b+2^{-1} a(n_0+1)] + \\
&+ \sum_{j=1}^{n_0-1} \frac{A_j^{(0)}}{1-A_j^{(1)}} \prod_{l=j+1}^{n_0} \frac{A_l^{(2)}}{1-A_l^{(1)}} + \frac{A_{n_0}^{(0)}}{1-A_{n_0}^{(1)}} + \prod_{l=1}^{n_0} \frac{A_l^{(2)}}{1-A_l^{(1)}} ; \forall k > n_0
\end{aligned} \tag{I.5}$$

If $E\{I|u,d\}$ denotes the expected length of a CRI, given that the length of the examined interval is u and the lag is d , then,

$$E\{I|u,d\} = \sum_{k=0}^{\infty} L_{k,0} e^{-\lambda u} \frac{(\lambda u)^k}{k!} \tag{I.6}$$

The bounds in (I.5) are used in the derivation of upper bounds on the expected value in (I.6). The largest n_0 in (I.5) is selected, the tighter those bounds are. Lower bounds on (I.6) are derived by truncation of the system.

Bounds on the Quantities W and Z

Let W and Z be as in (3.3), and for the sequence $\{T_i\}$ being as in Section 3.2, let us define,

$$H = E\{T_2 - T_1\} \tag{I.7}$$

For the computation of the expected values W and Z , we also need the computation of the expected value H in (I.7). Towards that, let us define the following quantities:

- n_u : The number of packet arrivals in an examined interval whose length is u .
- z_u : The sum of delays of the n_u packets, after the beginning of the CRI.
- ψ_u : The sum of delays of the n_u packets, before the beginning of the CRI.
- l_u : The number of slots needed to resolve an examined interval whose length is u .
- h_d : The number of slots needed to return to lag equal to one when starting from a collision resolution instant with lag d .
- w_d : The cumulative delay experienced by all the packets that were successfully transmitted during the h_d slots.
- $P(l|u)$: Given that the examined interval has length u , the probability that the corresponding collision resolution interval has length l .

$$H_d = E\{h_d\} \quad (I.8)$$

$$W_d = E\{w_d\}$$

We note that $H=H_1$, $W=W_1$, and $Z \approx \lambda H$. The following recursions are induced by the algorithm.

$$1 \leq d \leq \Delta ; h_d = \begin{cases} 1 & ; \text{if } l_d = 1 \\ l_d + h_{l_d} & ; \text{if } l_d > 1 \end{cases}$$

$$d > \Delta ; h_d = l_\Delta + h_{d-\Delta+l_\Delta} \quad (I.9)$$

$$1 \leq d \leq \Delta; w_d = \begin{cases} \psi_d + z_d; & \text{if } l_d = 1 \\ \psi_d + z_d + w_{l_d}; & \text{if } l_d > 1 \end{cases}$$

$$d > \Delta; w_d = \psi_\Delta + z_\Delta + (d - \Delta)n_\Delta + w_{d - \Delta + l_\Delta}$$

The above recursions yield the following infinite dimensionality linear systems:

$$H_d = \begin{cases} E\{l_d\} + \sum_{l=2}^{\infty} H_l P(l|d) & ; 1 \leq d \leq \Delta \\ E\{l_\Delta\} + \sum_{l=1}^{\infty} H_{d-\Delta+l} P(l|\Delta) & ; d > \Delta \end{cases} \quad (I.10)$$

$$W_d = \begin{cases} E\{\psi_d + z_d\} + \sum_{l=2}^{\infty} W_l P(l|d) & ; 1 \leq d \leq \Delta \\ E\{\psi_\Delta + z_\Delta + (d - \Delta)n_\Delta\} + \sum_{l=1}^{\infty} W_{d-\Delta+l} P(l|\Delta) & ; d > \Delta \end{cases} \quad (I.11)$$

; where, for Poisson traffic intensity λ , we have:

$$E\{l_u\} = \sum_{k=0}^{\infty} e^{-\lambda u} \frac{(\lambda u)^k}{k!} L_{k,0} \quad (I.12)$$

Also, if $P_k(l)$ denotes the probability that a multiplicity k collision is resolved in l slots, and if $l_{k,m}$ denotes the number of slots from a multiplicity k collision to the first successful transmission, given k packets with counter values 1 and m packets with counter values 2, then,

$$P(l_{k,m} = 0) = 0; \forall k, m$$

$$P(l_{0,m} = s) = \begin{cases} P_m, & s = 2 \\ 2^{-m}(1-P_m) \sum_{i=0}^m \binom{m}{i} P(l_{i,m-i} = s-2), & s \geq 3 \end{cases} \quad (I.13)$$

$$k \geq 1; P(l_{k,m} = s) = \begin{cases} P_k, & s = 1 \\ \sum_{i=0}^k \binom{k}{i} 2^{-k}(1-P_k) P(l_{i,k+m-i} = s-1), & s \geq 2 \end{cases}$$

$$P_0(l) = \begin{cases} 1, & l = 1 \\ 0, & \text{otherwise} \end{cases}$$

(I.14)

$$k \geq 1; P_k(l) = \sum_{s \geq 1} P(l_{k,0} = s) P_{k-1}(l-s)$$

Upper and lower bounds on the expected values in (I.10) and (I.11) are found via the methodology in [6]; thus, further details are omitted here. Those bounds are functions of the parameters p and q , used in the sequence $\{P_k\}_{k \geq 1}$ of capture probabilities.

Appendix II

Analysis of Algorithm for Simple Ternary Feedback

For the computation of the expected values W , Z , and H , in (4.7), we need the following quantities:

- n_u : The number of packet arrivals in an examined interval of length u , that are successfully transmitted during the collision resolution process.
- z_u : The sum of delays of the n_u packets, after the beginning of the CRI.
- Z_{k_1, k_2} : Given k_i , $i=1,2$, packets with counter values equal to i , $i=1,2$, the expected sum of the delays of those that are successfully transmitted by the algorithm.
- $\Psi_{u,d}$: The sum of delays of the n_u packets, before the beginning of the CRI, given that the lag at the beginning of the CRI equals d .
- l_u : The number of slots needed to resolve an examined interval whose length is u .
- h_d : The number of slots needed to return to lag equal to one when starting from a collision resolution instant with lag d .
- w_d : The cumulative delay experienced by all the packets that were successfully transmitted during the h_d slots.
- α_d : The number of packets that are successfully transmitted within the interval that corresponds to h_d .

$P(l|u)$: Given that the examined interval has length u , the probability that the corresponding collision resolution interval has length l .

$$\begin{aligned} H_d &= E\{h_d\} \\ W_d &= E\{w_d\} \\ A_d &= E\{\alpha_d\} \end{aligned} \quad (II.1)$$

We note that $H = H_1$, $W = W_1$, and $Z = A_1$. Also, for Poisson traffic with intensity λ , and for $L_{n,k-n}^{(m)}$ and $N_{n,k-n}^{(m)}$ as in Section 4.2, we have (for $P_k = pq^{k-1}$):

$$E\{l_u\} = \sum_{k \geq 0} e^{-\lambda u} \frac{(\lambda u)^k}{k!} L_k^{(m)} = \sum_{k \geq 1} e^{-\lambda u} \frac{(\lambda u)^k}{k!} L_{k,0}^{(m)} + \left[1 + \frac{p}{q}\right] e^{-\lambda u} - \frac{p}{q} e^{-\lambda u(1-q)} \quad (II.2)$$

$$E\{n_u\} = \sum_{k \geq 0} e^{-\lambda u} \frac{(\lambda u)^k}{k!} [k - N_k^{(m)}] = \lambda u - \sum_{k \geq 1} e^{-\lambda u} \frac{(\lambda u)^k}{k!} N_k^{(m)} \quad (II.3)$$

$$E\{z_u\} = \sum_{k=0}^{\infty} e^{-\lambda u} \frac{(\lambda u)^k}{k!} Z_{k,0} \quad (II.4)$$

In addition,

$$E\{\Psi_{u,d}\} = \begin{cases} 2^{-1} u E\{n_u\} & ; 1 \leq d \leq \Delta \\ [d - 2^{-1} u] E\{n_u\} & ; d > \Delta \end{cases} \quad (II.5)$$

The following recursions are induced by the algorithm:

$$1 \leq d \leq \Delta ; h_d = \begin{cases} 1 & ; \text{if } l_d = 1 \\ l_d + h_{l_d} & ; \text{if } l_d > 1 \end{cases} \quad (II.6)$$

$$d > \Delta ; h_d = l_\Delta + h_{d-\Delta+l_\Delta}$$

$$1 \leq d \leq \Delta ; w_d = \begin{cases} \Psi_{d,d} + z_d & ; \text{if } l_d = 1 \\ \Psi_{d,d} + z_d + w_{l_d} & ; \text{if } l_d > 1 \end{cases} \quad (II.7)$$

$$d > \Delta ; w_d = \Psi_{\Delta,d} + z_\Delta + w_{d-\Delta+l_\Delta}$$

$$1 \leq d \leq \Delta; \alpha_d = \begin{cases} n_d & ; \text{if } l_d = 1 \\ n_d + \alpha_{l_d} & ; \text{if } l_d > 1 \end{cases} \quad (\text{II.8})$$

$$d > \Delta; \alpha_d = n_\Delta + \alpha_{d-\Delta+l_d}$$

$$\begin{aligned} k_1 \geq 1; Z_{k_1, k_2} = P_{k_1} & \left\{ X_{k_1}^{(m)} \left[1 + \frac{m+1}{2} \right] + (m+1) \left[k_2 - N_{k_2}^{(m)} \right] + Z_{k_2, 0} \right. \\ & \left. + \left[1 - P_{k_1} \right] \left[k_1 + k_2 - N_{k_1, k_2}^{(m)} \right] + 2^{-k_1} \left[1 - P_{k_1} \right] \sum_{i=0}^{k_1} \binom{k_1}{i} Z_{i, k_1+k_2-i} \right\} \end{aligned} \quad (\text{II.9})$$

$$Z_{0, k} = k - N_k^{(m)} + Z_{k, 0} \quad (\text{II.10})$$

The above recursions yield the following infinite dimensionality linear systems:

$$H_d = \begin{cases} E\{l_d\} + \sum_{l=2}^{\infty} H_l P(l|d) & ; 1 \leq d \leq \Delta \\ E\{l_\Delta\} + \sum_{l=1}^{\infty} H_{d-\Delta+l} P(l|\Delta) & ; d > \Delta \end{cases} \quad (\text{II.11})$$

$$W_d = \begin{cases} E\{\Psi_{d,d} + z_d\} + \sum_{l=2}^{\infty} W_l P(l|d) & ; 1 \leq d \leq \Delta \\ E\{\Psi_{\Delta,d} + z_\Delta\} + \sum_{l=1}^{\infty} W_{d-\Delta+l} P(l|\Delta) & ; d > \Delta \end{cases} \quad (\text{II.12})$$

$$A_d = \begin{cases} E\{n_d\} + \sum_{l=2}^{\infty} A_l P(l|d) & ; 1 \leq d \leq \Delta \\ E\{n_\Delta\} + \sum_{l=1}^{\infty} A_{d-\Delta+l} P(l|\Delta) & ; d > \Delta \end{cases} \quad (\text{II.13})$$

Let (k, n) denote the state where k packets have counter value equal to 1 and n packets have counter value equal to 2. Let $l_{k,n}^{(m)}$ denote the number of slots needed by the algorithm to go from state (k, n) to state $(0, 0)$, given that the algorithm utilizes the integer m in Steps 3.a and 3.b of its operation. Then, the following recursions are induced,

where $P()$ means probability.

$$P(l_{k,n}^{(m)}=0) = 0 ; \forall k,n , P(l_{0,0}^{(m)}=1) = 1 \quad (\text{II.14})$$

$$n \geq 1 ; P\left[l_{0,n}^{(m)}=s\right] = \begin{cases} P_n ; s=m+2 \\ 2^{-n}(1-P_n) \sum_{i=0}^n \binom{n}{i} P\left(l_{i,n-i}^{(m)}=s-2\right) ; s>m+2 \end{cases} \quad (\text{II.15})$$

$$\left. \begin{matrix} k \geq 1 \\ n \geq 0 \end{matrix} \right\} ; P\left[l_{k,n}^{(m)}=s\right] = \begin{cases} P_k ; s=m+1 \\ 2^{-k}(1-P_k) \sum_{i=0}^k \binom{k}{i} P\left[l_{i,k+n-i}^{(m)}=s-1\right] ; s>m+1 \end{cases} \quad (\text{II.16})$$

In addition, given Poisson traffic intensity λ , we have:

$$P(l|u) = \sum_{k=0}^{\infty} e^{-\lambda u} \frac{(\lambda u)^k}{k!} P(l_{k,0}^{(m)} = l) \quad (\text{II.17})$$

Bounds

Let us define the sets $\{B_n\}_{n \geq 1}$, $\{A_n^{(j)}, 0 \leq j \leq n\}_{n \geq 1}$, $\{C_n\}_{n \geq 1}$, and $\{D_n^{(j)}, 0 \leq j \leq n\}_{n \geq 1}$, as follows:

$$B_1 \triangleq (1+P_1)^{-1} [3+(2m-1)P_1] \quad (\text{II.18})$$

$$B_n \triangleq [1-2^{-n}(1-P_n)]^{-1} \left\{ 1+mP_n+2^{-n}(1-P_n) \left[1+\sum_{i=1}^{n-1} \binom{n}{i} B_i \right] \right\}, n \geq 2$$

$$A_1^{(0)} \triangleq (1+P_1)^{-1} (1-P_1)$$

$$A_n^{(0)} \triangleq [1-2^{-n}(1-P_n)]^{-1} 2^{-n}(1-P_n) \left[1+\sum_{i=1}^{n-1} \binom{n}{i} A_i^{(0)} \right], n \geq 2 \quad (\text{II.19})$$

$$A_n^{(j)} \triangleq [1-2^{-n}(1-P_n)]^{-1} 2^{-n}(1-P_n) \sum_{i=j}^{n-1} \binom{n}{i} A_i^{(j)}, 1 \leq j \leq n-1, n \geq 2$$

$$A_n^{(n)} \triangleq [1-2^{-n}(1-P_n)]^{-1} P_n, n \geq 1$$

$$C_1 \triangleq 0, C_2 \triangleq (1+P_2)^{-1} 2P_2[2-X_2^{(m)}] \quad (\text{II.20})$$

$$C_n \triangleq [1-2^{-n}(1-P_n)]^{-1} \left\{ P_n[n-X_n^{(m)}] + 2^{-n}(1-P_n) \sum_{i=2}^{n-1} \binom{n}{i} C_i \right\}, n \geq 3$$

$$D^{(0)} \triangleq (1+P_1)^{-1} (1-P_1)$$

$$D_n^{(0)} \triangleq [1-2^{-n}(1-P_n)]^{-1} 2^{-n}(1-P_n) \left[1 + \sum_{i=1}^{n-1} \binom{n}{i} D_i^{(0)} \right], n \geq 2 \quad (\text{II.21})$$

$$D_n^{(j)} \triangleq [1-2^{-n}(1-P_n)]^{-1} 2^{-n}(1-P_n) \sum_{i=j}^{n-1} \binom{n}{i} D_i^{(j)}, 1 \leq j \leq n-1, n \geq 2$$

$$D_n^{(n)} \triangleq [1-2^{-n}(1-P_n)]^{-1} P_n$$

It can be found by induction, that there exist natural numbers n_0 and k_0 , such that:

$$b_l + a_l n < B_n < b_u + a_u n \quad ; \forall n > n_0 \quad (\text{II.22})$$

$$c_l^{(j)} < A_n^{(j)} < c_u^{(j)}, 0 \leq j \leq n$$

$$d_l < C_n < d_u$$

$$; \forall n > k_0 \quad (\text{II.23})$$

$$e_l^{(j)} < D_n^{(j)} < e_u^{(j)}, 0 \leq j \leq n$$

It is also found then, that:

$$L_{n,k-n}^{(m)} = B_n + \sum_{i=0}^n A_n^{(i)} L_{k-i,0}^{(m)}$$

and thus,

$$L_{1,0}^{(m)} = [1-A^{(0)}]^{-1} [B_1 + A^{(1)}]$$

$$(\text{II.24})$$

$$L_{k,0}^{(m)} = [1 - A_k^{(0)}]^{-1} \left\{ B_k + A_k^{(k)} + \sum_{i=1}^{k-1} A_k^{(i)} L_{k-i,0}^{(m)} \right\}$$

$$N_{n,k-n}^{(m)} = C_n + \sum_{i=0}^n D_n^{(i)} N_{k-i,0}^{(m)}$$

and thus,

$$N_{1,0}^{(m)} = 0, \quad N_{2,0}^{(m)} = (P_1 + P_2)^{-1} P_2 (1 + P_1) [2 - X_2^{(m)}] \quad (\text{II.25})$$

$$N_{k,0}^{(m)} = [1 - D_k^{(0)}]^{-1} \left\{ C_k + \sum_{i=1}^{k-2} D_k^{(i)} N_{k-i,0}^{(m)} \right\}, \quad k \geq 3$$

From (II.21) in conjunction with (II.23), and from (II.26) in conjunction with (II.24), we conclude that n_0 and k_0 can be found, such that there exist constants $A_u, A_l, B_u, B_l, C_u, C_l, D_u, D_l, F_u, F_l$ which satisfy the inequalities:

$$C_l + B_l k + A_l k^2 < L_{k,0}^{(m)} < A_u k^2 + B_u k + C_u; \quad \forall k > n_0 \quad (\text{II.27})$$

$$F_l + D_l k < N_{k,0}^{(m)} < D_u k + F_u; \quad \forall k > k_0 \quad (\text{II.28})$$

and the bounds in (II.27) and (II.28) are tight. We used those bounds to bound the quantities in (II.2) and (II.3).

Let us define,

$$E_n \triangleq [1 - 2^{-n}(1 - P_n)]^{-1} \left\{ P_n [X_n^{(m)}(1 + m/2) - m/2] + 2^{-n}(1 - P_n) \sum_{i=0}^{n-1} \binom{n}{i} E_i \right\}; \quad n \geq 1 \quad (\text{II.29})$$

$$F_n^{(j)} \triangleq \begin{cases} P_n [1 - 2^{-n}(1 - P_n)]^{-1} & ; j = n \\ [1 - 2^{-n}(1 - P_n)]^{-1} 2^{-n}(1 - P_n) \sum_{i=j}^{n-1} \binom{n}{i} F_i^{(j)} & ; 0 \leq j \leq n-1 \end{cases} \quad (\text{II.30})$$

$$G_n^{(j)} \Delta \begin{cases} (m+1)P_n [1-2^{-n} (1-P_n)]^{-1} & ; j=n \\ [1-2^{-n}(1-P_n)]^{-1} 2^{-n}(1-P_n) \sum_{i=j}^{n-1} \binom{n}{i} G_i^{(j)} & ; 0 \leq j \leq n-1 \end{cases} \quad (\text{II.31})$$

Then, from (II.9), (II.29), (II.30), and (II.31), we find:

$$Z_{n,k-n} = E_n + \sum_{i=0}^{n-1} F_n^{(i)} Z_{k-i,0} + \sum_{j=0}^{n-1} G_n^{(j)} [k-N_{k-i,0}] \quad (\text{II.32})$$

Also, from (II.29) - (II.31), and by induction, we find that there exist constants $f_l, e_l, f_u, e_u, g_l, g_u, h_l$, and h_u , and some n_0 , such that:

$$\begin{aligned} f_l + e_l n &\leq E_n \leq f_u + e_u n ; \forall n \geq n_0 \\ g_l &\leq F_n^{(j)} \leq g_u ; 0 \leq j \leq n, \forall n \geq n_0 \\ h_l &\leq G_n^{(j)} \leq h_u ; 0 \leq j \leq n, \forall n \geq n_0 \end{aligned} \quad (\text{II.33})$$

Substituting the bounds in (II.33), in expression (II.32), we find:

$$f_l + (g_l + h_l) k + e_l k^2 \leq Z_{k,0} \leq f_u + (g_u + h_u) k + e_u k^2 ; \forall k \geq n_0 \quad (\text{II.34})$$

We used the bounds in (II.34) to compute bounds on the expected value in (II.4).

DISTRIBUTION LIST

Copy No.

- 1 - 6 Director
Naval Research Laboratory
Washington, DC 20375

Attention: Code 2627
- 7 Dr. R. N. Madan
Electronics Division, Code 1114S3
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217-5000
- 8 Mr. James G. Smith
Office of Naval Research
Code 1241
800 N. Quincy Street
Arlington, VA 22217-5000
- 9 Professor Mike Athans
MIT, Building 35
Cambridge, MA 02139
- 10 Professor A. Ephremides
Electrical Engineering Dept.
University of Maryland
College Park, MD 20742
- 11 Dr. Dave Castanon
ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803-4901
- 12 Dr. John P. Lehoczky
Department of Statistics
Carnegie Mellon University
Schenley Park
Pittsburgh, PA 15213-3890
- 13 Professor Alex Levis
MIT, Building 35
Cambridge, MA 02139
- 14 Dr. Harold L. Hawkins
Manager, Perceptual Sciences
Code 1142
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

(Continued on Next Page)

- 15 Dr. G. Malecki
 Office of Naval Research
 Code 1142
 800 N. Quincy Street
 Arlington, VA 22217-5000
- 16 Dr. David Kleinman
 ESE Department, U-157
 The University of Connecticut
 STORRS, CT 06268
- 17 - 28 Defense Technical Information Center, S47031
 Building 5, Cameron Station
 Alexandria, VA 22314
- 29 - 30 P. Kazakos, EE
- 31 D. F. Lyons
- 32 - 33 E. H. Pancake, Clark Hall
- 34 SEAS Publications Files
- * Office of Naval Research Resident
 Representative
 818 Connecticut Avenue, N. W.
 Eighth Floor
 Washington, D.C. 20006
- Attention: Mr. Michael McCracken
 Administrative Contracting
 Officer

*Send Cover Letter Only

END

DATE

FILMED

8-88

DTIC